НАСТАВА РАЧУНАРСТВА

Др Драган Урошевић

СТРУКТУРЕ ПОДАТАКА ЗА ЕФИКАСНО ПРЕБРОЈАВАЊЕ НАД СКУПОВИМА ТАЧАКА

1. Увод

У овом тексту ћемо описати структуре података које се могу користити за репрезентовање колекције (скупа) тачака у неком d-димензионом ($d \geqslant 2$) простору тако да се може ефикасно одговарати на упите типа колико тачака из дате колекције припада неком хиперкубу у том простору.

Свака тачка из колекције је одређена својим координатама које образују уређену d-торку $(x_1, x_2, \ldots, x_d)$. Сматраћемо да су све координате реални бројеви. Хиперкуб је скуп тачака у d-димензионом простору одређен координатама два наспрамна темена: $a^L : (x_1^L, x_2^L, \ldots, x_d^L)$ и $a^U : (x_1^U, x_2^U, \ldots, x_d^U)$, а чине га све тачке тог простора чије су координате између одговарајућих координата та два темена:

$$H(a^L, a^U) = \{(x_1, x_2, \ldots, x_d) : x_i^L \leqslant x_i \leqslant x_i^U, i = 1, 2, \ldots, d\}.$$

Лако се закључује да је за $d = 2$ простор коме припадају те тачке могуће представити помоћу координатне равни (Декартов кординатни систем), а да хиперкубови представљају правоугаонике чије су странице паралелне осама и који су одређени паровима наспрамних темена. Ако је $d = 3$, хиперкубови су квадри у тродимензионом простору (који истина не можемо баш тако лако представити графички, али можемо замислити).

Претпоставимо да имамо неку унапред задату колекцију тачака S и желимо да поставимо серију упита Q облика: колико тачака из скупа S припада хиперкубу чија су два наспрамна темена тачке a_j^L и a_j^U ($j = 1, 2, \ldots, |Q|$). На први поглед, ово је тривијалан проблем јер се за сваку тачку релативно лако проверава да ли припада задатом хиперкубу (проверавајући за сваку њену координату да ли је између одговарајућих координата тачака – темена која одређују хиперкуб). Провером за сваку тачку да ли је у хиперкубу и бројањем само оних које јесу у хиперкубу, релативно лако долазимо до одговора на упит. Међутим, ако је n број тачака у скупу и q број упита, лако закључујемо да одговор на све упите захтева $\Theta(nq)$ операција (то су углавном поређења бројева – координата тачака) и сложићете се да то са повећањем броја упита може представљати не баш најбоље решење.

Због тога се наметала потреба проналажења начина за репрезентовање дате колекције на рачунарима, тако да се смањи сложеност одређивања одговора на описане упите. Тако су настале две структуре за репрезентовање колекције тачака над којом се изводе упити: k-d стабла и R-стабла.

Напоменимо на почетку да се примена k-d стабала и R стабала не своди само на проблем бројања тачака које припадају некој конкретној области (хиперкубу). Уз незнатне варијације (модификације и/или допуне) исте структуре података могу бити искоришћене за решавање следећих пролема (задатака, упита):

- одређивање најближе тачке задатој тачки, што може бити занимљиво за разне практичне примене: у мобилној телефонији, у организацији ватрогасне службе, служби хитне помоћи (за одређивање најближе ватрогасне станице, односно најближе станице хитне помоћи месту на коме је потребна њихова услуга, тј. месту на коме се десила нека несрећа);
- одређивање највеће „празне“ области (тј. највећег хиперкуба који не садржи ни једну тачку) или највеће „скоро празне“ области (тј. највећег хиперкуба који садржи не више од довољно мало тачака) – може бити корисно при планирању градње нових објеката на простору који садржи одређене препреке;
- при решавању оптимизационих проблема као што су Проблем Трговачког Путника, Проблем Рутирања Возила (обезбеђују да се на ефикасан начин одреди како наставити маршруту трговачког путника, односно маршруту возила).

2. k-d стабло

2.1. Опис структуре

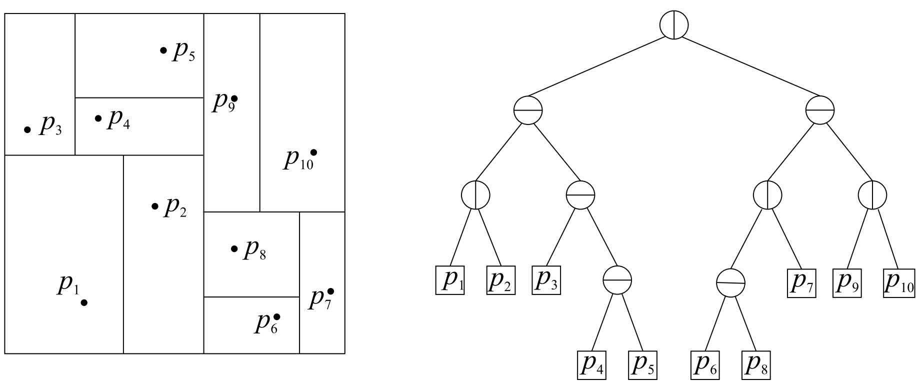
k-d стабло представља варијанту бинарног стабла. Стабло је састављено од чворова који могу бити *интерни чворови* и *чворови-листови*. Сваки интерни чвор стабла дели простор (или неки подскуп целог d-димензионог простора) на два, раздвајајући тачке тог простора на основу вредности једне од њихових координата. У сваком интерном чвору u стабла се налазе следећи подаци:

- $u.gr\_dim$ – ознака (редни број) координате по којој се врши раздвајање тачака (цео број између 1 и d);
- $u.gr\_vred$ – вредност на основу које се врши дељење простора (раздвајање тачака);
- $u.br\_t$ – број тачака смештених у подстаблу чији је корен тај чвор.

Сваки интерни чвор има два наследника (односно две гране које воде ка наследницима). Та два наследника су корени два подстабла. Леви наследник је корен подстабла у коме су смештене све тачке из скупа тачака које су смештене у подстаблу чији је корен u за које важи да је координата са редним бројем $u.gr\_dim$ мања или једнака од $u.gr\_vred$. Десни наследник је корен подстабла у коме су смештене тачке чија је координата са редним броје $u.gr\_dim$ већа од $u.gr\_vred$.

Ако вредност за гранање (деобу тачака) изаберемо тако да приближно половину тачака треба сместити у свако од подстабала, онда ће се број тачака које смештамо у подстабла смањивати те ће на крају остати само по једна тачка. Тада се та тачка смешта у чвор типа *чвор–лист* који представља комплетно стабло. Једини податак који се чува у таквом чвору је тачка ($u.tacka$) која је сачувана у њему, односно координате те тачке.

На слици 1 је приказан један од могућих начина да се репрезентује 10 нацртаних тачака. Ради поједностављења, интерни чворови су приказани кружићима. Исцртани пречник визуелно представља координату по којој се врши дељење простора (вертикални, ако се дељење врши по x координати, а хоризонтални, ако се дељење врши по y координати).



Сл. 1. Један од могућих начина да се задате тачке сместе у k-d стабла

Остало је неколико детаља који нису прецизирани кроз претходни опис. Први је начин одређивања координате по којој се врши дељење. Постоји неколико правила која се најчешће примењују:

- циклична промена координате по којој се врши дељење: координата за дељење у корену стабла је 1, у његовим наследницима 2, у њиховим наследницима 3, итд.
- дељење се на сваком нивоу обавља по оној координати за коју је разлика између највеће и најмање вредности те координате максимална (највећа), односно по координати по којој је распон вредности координате највећи.
- на сваком нивоу се на случајан начин бира координата по којој ће се извести дељење.

Када је одређена координата по којој се врши дељење простора, онда се за скуп вредности те координате одреди медијана и то ће бити вредност на основу које ће се вршити дељење простора (односно раздвајање тачака). Прецизније, ако је X_{gr} скуп састављен само од одговарајућих координата свих тачака смештених у том подстаблу, онда је медијана тог скупа број x_{med} са особином да је број елмената скупа X_{gr} мањих од x_{med} једнак броју елемената већих до x_{med}.

Ако је број елемената у скупу X_{gr} непаран, онда се за x_{med} може узети „средњи“ елемент по вредности. Ако је број елемената у скупу X_{gr} паран, онда се за x_{med} може узети аритметичка средина два „средња“ елемента. Ово све под претпоставком да су сви елементи скупа чију медијану одређујемо различити. Ако се неке вредности понављају, неформални опис се мало компликује (али мислимо да не треба томе да посвећујемо више пажње, јер то није главна тема овог чланка).

За крај овог описа одредимо величину меморијског простора потребног за k-d стабло за скуп од n тачака. Свака тачка треба да буде смештена у листу стабла, те према томе стабло има n листова. Како је у питању бинарно стабло, оно ће имати још $n-1$ интерних чворова. Укупан број чворова је $2n-1$, а у сваком чвору се чува ограничена количина информација те је укупан меморијски простор $O(n)$.

2.2. Формирање стабла

Функција за формирање k-d стабла за задати скуп тачака се може рализовати као рекурзивна функција. Након избора координате по којој ће се тачке тог скупа поделити и одређивања вредности за дељење тачака, тачке се поделе у два подскупа. Затим се рекурзивно формирају стабла за та два скупа тачака. Резултати извршавања су подстабла, а корени тих подстабала су леви и десни наледник корена целог стабла.

Функција за формирање k-d стабла је приказана у Алгоритму 1. Са $Pr_{u.gr\_dim}(t)$ је означена функција која одређује једну ($u.gr\_dim$-ту) координату тачке t.

Алгоритам 1. Алгоритам за формирање k-d стабла

FormirajStablo (S)
if $|S| = 1$ **then**
  $w \leftarrow$ novi cvor–list
  $w.taqka \leftarrow t$ (*t је једини елемент скупа S*)
  return w
$u \leftarrow$ novi cvor–interni
$u.gr\_dim \leftarrow OdrediKoordinatu$
$u.gr\_vred \leftarrow Medijana(S, u.gr\_dim)$
$S_L \leftarrow \{t \in S : Pr_{u.gr\_dim}(t) \leqslant u.gr\_vred\}$
$S_G \leftarrow \{t \in S : Pr_{u.gr\_dim}(t) > u.gr\_vred\}$
$u.left \leftarrow FormirajStablo(S_L)$
$u.right \leftarrow FormirajStablo(S_G)$
return u

Одредимо сложеност функције за формирање k-d стабла за скуп од n тачака. То ћемо извести анализирајући сложеност појединих корака у алгоритму:

- одређивање медијане за скуп од n елемената има сложеност $\Theta(n)$ (детаље можете пронаћи у [2];
- формирање корена стабла има сложеност $\Theta(1)$;

- након тога преостаје да се формирају k-d стабла за два подскупа који имају по $n/2$ елемената (ако је n непаран онда ће један подскуп имати $(n+1)/2$, а други $(n-1)/2$ тачака).

Ако са $T(n)$ означимо сложеност формирања k-d стабла за скуп који има n тачака имаћемо везу

$$T(n) = 2T(n/2) + g(n)$$

где је $g(n) = \Theta(n)$. По *Мастер теореми* (о којој више можете прочитати у [2]) добијамо да је $T(n) = \Theta(n \log n)$.

2.3. Одговор на упит

Приметимо да сваком чвору k-d стабла можемо придружити један хиперкуб – хиперкуб који садржи све тачке сачуване у подстаблу чији је корен баш тај чвор. Тај хиперкуб не мора бити јединствен. Између осталог може бити одређен паром темена (тачака) чије се координате рачунају као минимум, односно максимум одговарајућих координата свих тачака из тог подстабла.

Али, ако одредимо хиперкуб (односно његова два наспрамна темена) за корен целог стабла, онда се поступак одређивање за остале чворове може извести и на следећи начин:

- ради скраћења записа, уместо $u.gr\_dim$ користимо ознаку g, а уместо $u.gr\_vred$ ознаку x_g;
- нека су наспрамна темена хиперкуба који је придружен чвору u тачке
$$a^L : (x_1^L, x_2^L, \ldots, x_{g-1}^L, x_g^L, x_{g+1}^L, \ldots, x_d^L)$$
и
$$a^U : (x_1^U, x_2^U, \ldots, x_{g-1}^U, x_g^U, x_{g+1}^U, \ldots, x_d^U).$$
- Тада ће хиперкубови који одговарају подстаблима чвора u бити одређени следећим паровима темена:
  - за левог наследника $a_l^L : (x_1^L, x_2^L, \ldots, x_{g-1}^L, x_g^L, x_{g+1}^L, \ldots, x_d^L)$ и $a_l^U : (x_1^U, x_2^U, \ldots, x_{g-1}^U, x_g, x_{g+1}^U, \ldots, x_d^U)$;
  - за десног наследника $a_r^L : (x_1^L, x_2^L, \ldots, x_{g-1}^L, x_g, x_{g+1}^L, \ldots, x_d^L)$ и $a_r^U : (x_1^U, x_2^U, \ldots, x_{g-1}^U, x_g^U, x_{g+1}^U, \ldots, x_d^U)$.

Означимо са $H(u)$ хиперкуб придружен чвору u, односно подстаблу чији је корен чвор u. Ако треба да пребројимо колико тачака из стабла чији је корен чвор u припада неком хиперкубу H, онда морамо узети у обзир следеће чињенице:

- ако је u лист, онда треба проверити да ли тачка сачувана у том чвору припада или не хиперкубу H и у зависности од тога вратити резултат 1 или 0;
- ако је $H(u) \subset H$, онда све тачке из подстабла припадају хиперкубу H;
- ако је $H(u) \cap H = \emptyset$, онда ниједна тачка из подстабла не припада хиперкубу H;
- у свим осталим случајевима број тачака који припадају хиперкубу H одређујемо тако што одредимо број тачака за лево подстабло, десно подстабло и саберемо та два броја.

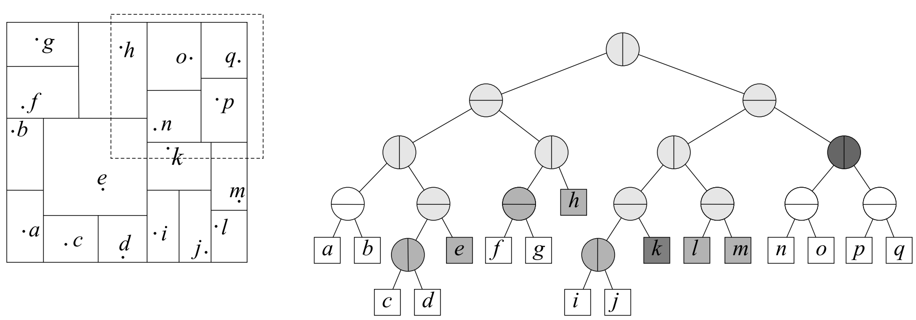
Функција за одређивање броја тачака који припадају хиперкубу је приказана у Алгоритму 2.

Алгоритам 2. Алгоритам за пребројавање тачака

PrebrojTachke (H, u)
if u je list **then**
  if $u.tacka \in H$ **then**
    return 1
  else
    return 0
if $H(u) \subset H$ **then**
  return $u.br\_t$
if $H(u) \cap H = \emptyset$ **then**
  return 0
return PrebrojTachke$(H, u.left)$ + PrebrojTachke$(H, u.right)$

Други случај у горњој анализи оправдава додавање податка $u.br\_t$ колекцији података који се чувају у интерним чворовима стабла.

На слици 2 је приказан пример k-d стабла и историја израчунавања броја тачака за правоугаоник приказан испрекиданим линијама. На k-d стаблу су необојени чворови који неће бити посећени током бројања. Најтамнијом нијансом сиве су обојени чворови у којима се прекида рекурзивно позивање, а резултат који одговара том чвору је управо број тачака сачуваних у том подстаблу. Нешто светлијом нијансом сиве су приказани чворови у којима се прекида поступак бројања зато што су хиперкуб који одговара том чвору и хиперкуб за који је постављен упит дисјунктни, или зато што је тај чвор лист, а тачка сачувана у том чвору не припада хиперкубу. Коначно, најсветлијом сивом су обојени чворови за које важи да ће функција за пребројавање бити позвана за његове наследнике.



Сл. 2. Илустрација за бројање за k-d стабла

Шта можемо рећи о сложености пребројавања? Ради поједностављења направимо анализу у дводимензионом случају тј. за $d = 2$. Приметимо да је сложеност одређена бројем чворова у стаблу кроз које треба проћи током пребројавања (тј. бројем чворова које треба посетити). Но да бисмо то одредили треба да проценимо колико има чворова са особином да ће бити посећен он и његова оба наследника (у остатку текста зваћемо такве чворове *проширени* чворови). То су заправо чворови са особином да се преклапају хиперкуб из упита и хиперкуб који одговара том чвору.

Ради поједностављења анализе претпоставимо да се раздвајање тачака врши наизменично по првој и другој координати (тј. по x и y координати). Ако је $d = 2$, хиперкуб је правоугаоник одређен по једним паром хоризонталних и вертикалних линија које садрже странице правоугаоника. За сваку од тих линија треба да проценимо колико хиперкубова одређених чворовима стабла пресеца та линија. Тиме ћемо добити и процену колико чворова ћемо обићи током пребројавања. Нека треба да проценимо за вертикалну линију и подстабло са n чворова колико има чворова чији хиперкуб пресеца та вертикална линија (означимо тај број са $C(n)$). Претпоставимо поред тога да се у корену стабла врши раздвајање по првој (x) координати.

Нека је u корен стабла и нека вертикална линија има једначину $x = x_0$. Зависно од тога да ли је $u.gr\_vred$ већа или мања од x_0, само хиперкубови који одговарају чворовима у левом или десном подстаблу ће имати заједничких тачака са вертикалном линијом. Претпоставимо да је $x_0 < u.gr\_vred$ (слично се ради ако је $x_0 > u.gr\_vred$). Очигледно да та вертикална линија пресеца и $H(u.left)$. Међутим, хиперкуб $H(u.left)$ се на следећем гранању дели хоризонталном линијом и вертикална линија $x = x_0$ пролази кроз оба новонастала хиперкуба (као и кроз неке хиперкубове који одговарају чворовима из оба подстабла левог подстабла). Зато укупан број правоугаоника који имају контакт са вертикалном линијом задовољава следећу неједнакост

$$C(n) \leqslant \begin{cases} 2, & \text{ако је } n \leqslant 4 \\ 2C(n/4) + 1, & \text{ако је } n > 4. \end{cases}$$

Користећи ту везу лако добијамо:

$$\begin{aligned} C(n) &\leqslant 1 + 2C\left(\frac{n}{4}\right) \\ &\leqslant 1 + 2\left(1 + 2C\left(\frac{n/4}{4}\right)\right) = (1+2) + 4C\left(\frac{n}{16}\right) \\ &\leqslant (1+2) + 4\left(1 + 2C\left(\frac{n/16}{4}\right)\right) = (1+2+4) + 8C\left(\frac{n}{64}\right) \\ &\leqslant \cdots \\ &\leqslant \sum_{i=0}^{k-1} 2^i + 2^k C\left(\frac{n}{4^k}\right) \end{aligned}$$

У последњој неједнакости k треба да буде такво да је $\frac{n}{4^k} \leqslant 1$, тј. $n \leqslant 4^k$, а из овога следи $k = \lceil \log_4 n \rceil$. Али, ради поједностављења рачуна узећемо да је $k = \log_4 n$. Тада ће бити

$$C(n) \leqslant 2^k - 1 + 2^k C(1) \leqslant 3 \cdot 2^k.$$

Како је

$$2^k = 2^{\log_4 n} = 2^{(\log_2 n)/2} = (2^{\log_2 n})^{1/2} = n^{1/2} = \sqrt{n},$$

то је $C(n) = O(\sqrt{n})$.

Проширени чворови су сви чворови којима одговарају хиперкубови (правоугаоници) који имају заједничких тачака са бар једном од правих одређених страницама хиперкуба (правоугаоника) из упита. Према томе, број проширених чворова је највише 4 пута већи од броја $C(n)$, тј. $O(\sqrt{n})$. Број посећених чворова је ограничен производом броја проширених чворова и коначног броја па је и тај број $O(\sqrt{n})$.

Показује се да је сложеност упита за произвољно d ($d \geqslant 2$) једнака $O(n^{\frac{d-1}{d}})$.

3. *R*-стабло

Сложеност упита за k-d стабло није завидна. Поготово када се повећа димензија простора коме припадају тачке које обрађујемо. Због тога је настала нова структура података, такозвано R-стабло.

Код R-стабала је искоришћена чињеница да се проблем бројања тачака у неком хиперкубу може представити као композиција два или више проблема бројања где ће се бројања вршити у простору мање димензије. Али, то је онда зехтевало и измене у структури података које ће то омогућити. Те измене су довело до повећање меморијских захтева.

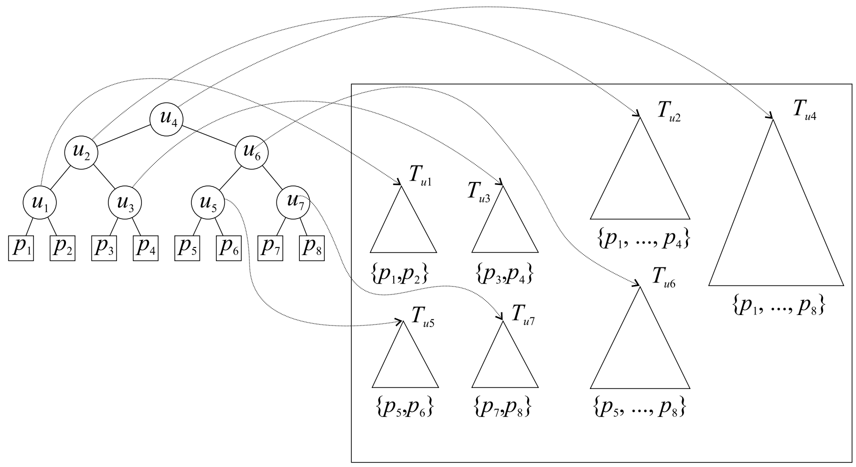
3.1. *R*-стабло за $d = 2$

R-табло је варијанта бинарног стабла у чијим листовима треба да буду сачуване тачке из колекције, при чему се оне распоређују у листове на основу вредности x–координате. Сваки интерни чвор u садржи као информацију један број $u.x_g$ на основу кога ће све тачке сачуване у подстаблу чији је корен u бити подељене у два подскупа:

- оне чија је x-координата мања од $u.x_g$ (биће сачуване у левом подстаблу);
- оне чија је x координата већа од x_g (биће сачуване у десном подстаблу).

Намеће се потреба да се за $u.x_g$ бира број такав да половина тачака буде смештено у лево подстабло, а друга половина у десно подстабло. Према томе, за $u.x_g$ треба узимати медијану скупа састављеног од x–координата тачака из скупа оних које су сачуване у подстаблу.

Поред тога за сваки чвор u се формира помоћно стабло у које се смештају све тачке из скупа оних сачуваних у подстаблу чији је корен u, али овај пут уређених по вредности y–координате. То значи да су тачке сачуване у листовима помоћног стабла, док су у осталим чворовима сачувани бројеви који имају улогу да раздвоје тачке које ће бити смештене у подстаблима стабла чији је корен тај чвор. Наравно, и у овом случају те вредности се бирају тако да помоћно стабло буде балансирано. Означићемо то помоћно стабло са T_u (али ћемо користити и ознаку $u.T$ у алгоритмима). На слици 3 је приказано како би изгледала цела структура за скуп који има осам тачака.



Сл. 3. Пример R-стабла

3.2. Меморијски захтеви

Ако је у стаблу смештено n тачака, онда основно стабло има n листова и укупно $2n-1$ чворова. Сваки чвор има коначан скуп информација те је укупан меморијски захтев за основно стабло $O(n)$. Основно стабло се формира тако да буде балансирано па због тога има $O(\log n)$ нивоа. Свака тачка може бити смештена у више помоћних стабала, али у највише једном помоћном стаблу на сваком од нивоа. Према томе, укупан број листова у помоћним стаблима која одговарају једном нивоу основног стабла је n, а укупан број чворова је највише $2n-1$. Значи укупни меморијски захтев је $O(n \log n)$.

3.3. Формирање стабла

Препоручује се да формирање основног стабла почне од листова, коме би претходило сортирање тачака по вредности x-координате. То је редослед у коме су тачке смештене у листовима слева надесно. Претпоставимо да смо већ обавили одређени део поступка за формирање. То значи да смо спајајући мање целине формирали одређен број подстабала основног стабла. Спајамо по два подстабла основног стабла формирајући веће. При том спајању најзахтевнији део је формирање помоћног стабла за корен новонасталог подстабла. Да би то одрадили потребно да се све тачке које припадају том подстаблу уреде по y-координати и након тога распореде у балансирано бинарно стабло. Међутим, два помоћна стабла која одговарају подстаблима основног стабла, која су спојена су већ оформљена и у њиховим листовима су тачке уређене по y-координатама тако да је довољно та два низа листова спојити у један сортирани низ. То се може извести у линеарном времену (по броју тачака). Према томе, формирање једног нивоа основног стабла има сложеност $O(n)$. Како има $O(\log n)$ нивоа, то је сложеност формирања стабла $O(n \log n)$.

3.4. Рачунање одговора на упит

Са сваким чвором u основног стабла асоцирана је вертикална трака, тако да све тачке смештене у том подстаблу припадају тој траци. Трака је ограничена са две вертикалне праве: $x = x_L(u)$ и $x = x_U(u)$, а чине је тачке чије су x-координате између $x_L(u)$ и $x_U(u)$. Означимо са $Tr_x(u) = [x_L(u), x_U(u)]$ траку која је асоцирана са чвором u.

За корен целог R-стабла можемо узети да је x_L једнака минималној вредности x-координата за скуп тачака за који се то стабло формира. Аналогно, за x_U се може узети максимална вредност x-координата. Алтернативно се за x_L и x_U могу узети неке вредности за које се може гарантовати да ниједна тачка неће имати мању, односно већу вредност x-координате.

Ако је са чвором u асоцирана трака $Tr_x(u) = [x_L(u), x_U(u)]$, онда је са његовим левим наследником асоцирана трака $[x_L(u), u.x\_g]$, а са десним трака $[u.x\_g, x_U(u)]$.

Слично је са сваким чвором v помоћних стабала асоцирана хоризонтална трака одређена паром правих $y = y_L(v)$ и $y = y_U(x)$, тако да све тачке сачуване у листовима подстабла чији је корен чвор v припадају тој траци (тј. њихове y-координате су између $y_L(v)$ и $y_U(v)$).

Сваки правоугаоник чије су странице паралелне осама представља Декартов производ два интервала $P = [x_L, x_U] \times [y_L, y_U]$, а важи

$$P = \{(x, y) | x_L \leqslant x \leqslant x_U, y_L \leqslant y \leqslant y_U\}$$

Бројање тачака које припадају неком правоугаонику почиње у корену целог стабла. За сваки чвор u који посетимо током тог бројања може наступити један од следећих случајева:

- интервали $[x_L, x_U]$ и $[x_L(u), x_U(u)]$ су дисјунктни, тада ниједна тачка из подстабла чији је корен u не може припадати правоугаонику P;
- интервал $[x_L(u), x_U(u)]$ је садржан у интервалу $[x_L, x_U]$ – тада све тачке из подстабла чији је корен u имају особину да су им x координате између x_L и x_U. Да би припадале правоугаонику P потребно је да и y координате буду између y_L и y_U. Али, да бисмо проверили колико има тачака које задовољавају то својство потребно је да бројање наставимо кроз помоћно стабло T_u. Тај део је врло сличан делу који описујемо уз једину разлику да када у помоћном стаблу дођемо до чвора v за који важи да је $[y_L(v), y_U(v)] \subset [y_L, y_U]$ онда све тачке подстабла чији је корен v припадају правоугаонику P.
- ако је $[x_L(u), x_U(u)] \cap [x_L, x_U] \neq \emptyset$ и $[x_L(u), x_U(u)] \not\subset [x_L, x_U]$, онда бројање морамо наставити у наследницима чвора u.

Алгоритам 3. Алгоритам за бројање тачака

$PrebrojTackeX(P = [x_L, x_U] \times [y_L, y_U], u)$
if u je list **then**
    if $u.tacka \in P$ **then**

```
        return 1
    else
        return 0
if [x_L, x_U] ∩ [x_L(u), x_U(u)] = ∅ then
    return 0
if [x_L(u), x_U(u)] ⊂ [x_L, x_U] then
    return PrebrojTackeY(P, root(u.T))
return PrebrojTackeX(P, u.left) + PrebrojTackeX(P, u.right)
```

Алгоритам 4. Алгоритам за бројање тачака

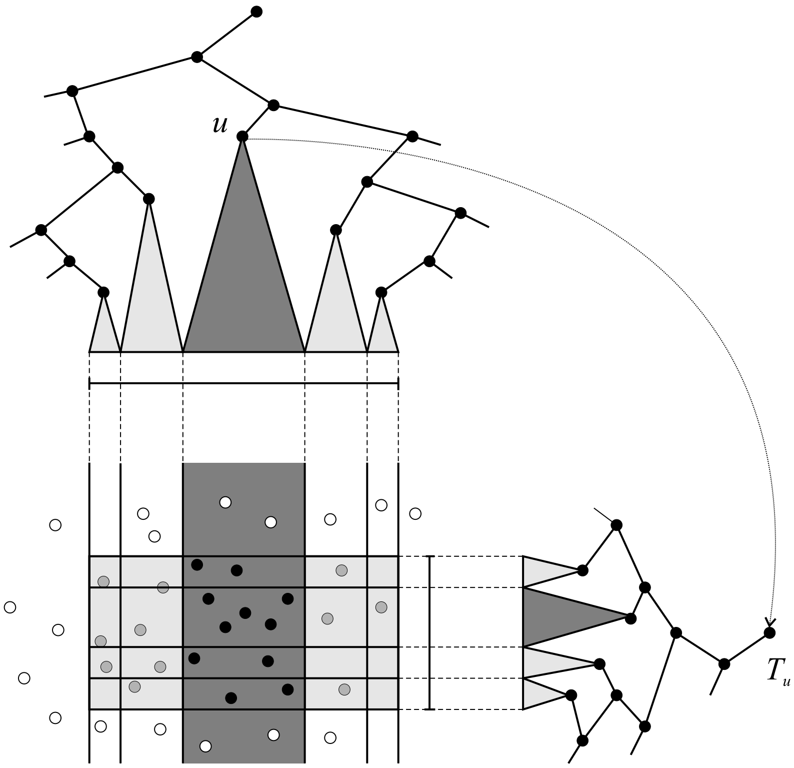
```
PrebrojTackeY(P = [x_L, x_U] × [y_L, y_U], u)
if u je list then
    if u.tacka ∈ P then
        return 1
    else
        return 0
if [y_L, y_U] ∩ [y_L(u), y_U(u)] = ∅ then
    return 0
if [y_L(u), y_U(u)] ⊂ [y_L, y_U] then
    return u.br_t
return PrebrojTackeY(P, u.left) + PrebrojTackeY(P, u.right)
```

На слици 4 је дата илустрација бројања. Шрафирана подстабла на слици 4 имају особине да траке које њима одговарају комплетно пролазе кроз правоугаоник P. Помоћно стабло које одговара корену средњег стабла (означимо тај чвор са u) приказано је са десне стране. Шрафирана подстабла у том стаблу имају особине да траке које њима одговарају пролазе кроз правоугаоник. Све тачке које припадају пресецима тих трака и вертикалне траке која одговара чвору u припадају правоугаонику (црно обојене тачке).

3.5. Анализа комплексности бројања

При бројању тачака интервали $[x_L, x_U]$ и $[y_L, y_U]$ се разбијају на унију *максималних* дисјунктних интервала који одговарају чворовима бинарних стабала. Ако се ти интервали бирају тако да буду што већи (максимални), онда их има највише по два на сваком нивоу бинарног стабла. Ако је бинарно стабло балансирано, онда стабло са n тачака има $O(\log n)$ нивоа, те ће тако бити $O(\log n)$ интервала у тој унији.

Покушајмо да укратко покажемо да је тако. Ако набројимо x-координате тачака смештених у листовима R-стабла слева на десно, онда ће тај низ бити сортиран у неопадајућем поретку. Нека је u крајње десни лист у коме је тачка са x-координатом мањом од x_L, а v крајње леви лист у коме је смештена тачка са x-координатом већом од x_U. Оправдано се поставља питање да ли такви листови постоје. Додавањем два листа у којима су смештене тачке са x-координатама $-\infty$ и $+\infty$, такви листови ће сигурно постојати.



Сл. 4. Илустрација бројања у R-стаблу

Сви листови између u и v садрже тачке чије су x-координате између x_L и x_U. До листова u и v постоје јединствени путеви од корена стабла. Како оба пута почињу у корену, они имају бар један заједнички чвор (баш корен). Нека је w чвор у којем се путеви раздвајају. Очигледно ће пут до u ићи преко левог наследника чвора w, а пут до v преко десног наследника. Ако пратимо пут до чвора u, сваки пут када у тренутном чвору w' скрећемо лево важи да комплетно десно подстабло стабла чији је корен w' припада интервалу $[x_L, x_U]$, а самим тим и интервал који одговара десном наследнику је подинтервал од $[x_L, x_U]$. За разлику од десног подстабла, неке од тачака левог подстабла стабла чији је корен чвор w' не припадају интервалу $[x_L, x_U]$ па самим тим ни интервал који одговара левом подстаблу није подинтервал од $[x_L, x_U]$. Ако пратимо пут до v, онда при скретању на десно важи да комплетно лево подстабло припада интервалу $[x_L, x_U]$, а самим тим и интервал који одговара левом подстаблу је подинтервал од $[x_L, x_U]$.

Према томе, при сваком померању дуж путу ка листовима u и v највише по један нови подинтервал може бити додат колекцији интервала, што значи да ће на сваком нивоу бити додата највише два подинтервала. Број подинтервала одређује колико ће бити рекурзивних позива у функцији за бројање тачака. Наиме са сваким чвором кроз који се пролази могу бити два рекурзивна позива: за левог и десног наследника. Ако пак интервал који одговара неком чвору представља подинтервал од $[x_L, x_U]$, онда се позива функција за бројање по помоћном стаблу. Како таквих чворова има $O(\log n)$ онда је толико и позива.

Потпуно исто разматрање важи и за бројање по помоћним стаблима, тј. сложеност бројања је $O(\log |T_u|)$, а како је $|T_u| < 2n-1$, то је сложеност $O(\log n)$. Према томе, сложеност комплетног бројања је $O(\log^2 n)$.

3.4. Шта ако је $d > 2$?

Ако је $d > 2$, онда се каскадно додају нова подстабла. Значи, основно стабло ће бити организовано на основу вредности прве координате. Сваки интерни чвор тог стабла ће делити простор на секторе у којима је ограничена вредност прве координате на неки интервал, док све остале координате узимају као вредности било који реални број. Сваком чвору u се придружује стабло $T_u^{(2)}$ које садржи тачке сачуване у подстаблу чији је корен u, овај пут организоване по вредности друге координате. Сваком чвору u' стабла $T_u^{(2)}$ се придружује стабло $T_{u'}^{(3)}$ које садржи тачке из подстабла чији је корен u' уређене по трећој координати. Наравно, поступак се наставља до последње координате.

Очигледно се меморијски захтеви повећавају те је величина потребног простора $O(n \log^{d-1} n)$.

Не би требало да буде тешко уопштити функцију за бројање тачака које припадају произвољном хиперкубу. Сложеност те функције ће бити $O(\log^d n)$.

4. Закључак

Такозвана претрага по области (назив за одређивање укупног броја тачака које пипадају некој области) је један од проблема који имају велику практичну примену.

Због тога је било неопходно развити структуре података које ће обезбедити ефикасно претрагу. Две овде описане се разликују по меморијским захтевима, али и по сложености претраге.

ЛИТЕРАТУРА

1. Bentley, J. L., *Multidimensional binary search trees used for associative searching*, Communications of the ACM, 18(9): 509-517, 1975.
2. Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., *Introduction to Algorithms*, Second Edition, MIT Press, 2001.
3. Goodman, J., O'Rourke, J., and Indyk, P. (Ed.) *Chapter 39: Nearest neighbours in high-dimensional spaces*, Handbook of Discrete and Computational Geometry (2nd ed.). CRC Press, 2004.
4. Guttman, A., *R-Trees: A Dynamic Index Structure for Spatial Searching*, Proceedings of the ACM SIGMOD international conference on Management of data - SIGMOD '84. pp. 47, 1984.

Математички институт САНУ, Кнеза Михаила 36, Београд

E–mail: draganu@mi.sanu.ac.rs