Universitatea Al.I.Cuza Iasi **Nume:**
Facultatea de Informatică **Grupa:**

Proiectarea algoritmilor – Test scris 11.04.2014, A

Observaţii:
1. Nu este permisă consultarea bibliografiei.
2. Toate întrebările sunt obligatorii.
3. Fiecare întrebare/item este notată cu un număr de puncte indicat în paranteză. Descrieți conceptele utilizate în răspunsuri.
4. Algorimii vor fi descriși în limbajul Alk. Se adimit extensii cu sintaxă inspirată din C++ (de exemplu, for, do-while, repeat-until, etc.). Pentru structurile de date utilizate se va preciza operațiile (fără implementare daca nu se cere explicit) și complexitățile timp și spațiu ale acestora.
5. Nu este permisă utilizarea de foi suplimentare.
6. Timp de răspuns: 1 oră.

1. Câtul împărțirii întregi a două numere întregi poate fi calculat prin scăderi repetate.
a) [0.5p] Să se formuleze problema împărțirii întregi, notată DIV, ca pereche (input, output).
b) [0.5p] Să se descrie un algoritm care rezolvă DIV.
c) [0.5p] Să se scrie configurațiile inițiale și finale și să se arate cum sunt reprezentate componentele perechii (input, output) de la a) în aceste configurații.
d) [0.5p] Să se arate ca algoritmul de la b) rezolvă corect problema descrisă la a).
e) [1p] Se consideră ca dimensiune a intrării numărul de biți *n* necesari reprezentării binare deîmpărțitului. Să se precizeze care este cazul cel mai nefavorabil pentru un *n* dat și care este complexitatea timp pentru acest caz (se va considera costul uniform).

Răspuns.
a) Input: a,b numere întregi.
Output: q = câtul împarțirii întregi a lui a la b, i.e. a = bq + r cu 0 <= r < |b|, daca b != 0, eroare daca b == 0.

b)
```
div(a, b) {
 if (b == 0) return "eroare";
 if (sign(a) + sign(b) == 0) s = -1;  // semne contrare
 else s = 1;          // acelasi semn
 if (a < 0 ) a = -a;  // se reduce la impartire
 if (b < 0) b = -b;  // de numere pozitive
 q = 0;
 while (a >= b) {
   a = a-b; q = q + 1;
 }
 retrun s*q;
}
```

functie care determina semnul
```
sing(x) {
 if (x < 0) returm -1;
 if (x > 0) returm 1;
 return 0;
}
```

c) configurația inițială: < div(a,b), a |-> a0 b |-> b0 >
configurația finală: < . , a |-> a1 b |-> b1 q |-> q0 s|-> s1 >, a0 = b0*q0 + a1, a1 < b
d) Relatia a < b este data de terminarea instructiunii while (imediat dupa terminare are loc intotdeauna negatia conditiei), iar la inceputul si la sfarsitul buclei while avem relatia a + b * q = |a0| (invariantul) deoarece a + b * q = (a – b) + b * (q + 1) .
e) Valoarea cea mai mare lui a reprezentat pe n biti este 2^n-1. Cazul cel mai negfavorabil este cand bucla while se executa de un numar maxim de ori, adica a este maxim, i.e. 2^n-1, si b minim, i.e., b =1. Bucla while se executa in acest caz de 2^n-1 ori, de unde rezulta complexitate timp este O(2^n).

Observatii:
1) S-a depunctat cu 0.25 daca nu s-a prevazut cazul b = 0.
2) S-au admis si justificari corecte bazate pe exemple (testare).
3) s-au admis si rezolvare corecta pentru cazul cand a,b >= 0, cu precizarea celelalte cazuri pot fi reduse la acesta.

2. a) [0.5p] descrieți problema CEI MAI APROPIAȚI VECINI.
b) [0.5p] Descrieți problema LOCUL GEOMETRIC AL CELOR MAI APROPIATE PUNCTE.
c) [1p] Să se arate că CEI MAI APROPIAȚI VECINI $\propto_n$ LOCUL GEOMETRIC AL CELOR MAI APROPIATE PUNCTE.
d) [0.5p] Să se arate că CEI MAI APROPIAȚI VECINI are complexitatea O(n log n). Justificați.
e) [0.5p] Ce se poate spune despre complexitatea Ω a problemei CEI MAI APROPIAȚI VECINI? Justificați.
Răspuns.
a) CEI MAI APROPIATI VECINI
Intrare: O multime S cu n puncte in plan.
Iesire: Cel mai apropiat vecin din S pentru fiecare punct din S.
Observatie: se recomanda de dat un exemplu pentru a dovedi ca s-a inteles problema de rezolvata.
b)
LOCUL GEOMETRIC AL CELOR MAI APROPIATE PUNCTE
Intrare: O mul¸time S cu n puncte in plan.
Ie¸sire: Pentru fiecare punct P din S, locul geometric la celor mai apropiate puncte, adica multimea punctelor mai apropiate de P decat de orice alt punct din S.
Observatie: Urmatorul text ajuta la explicatii (se putea utiliza si un exemplu in loc):
Presupunem $S = \{P_0, \ldots, P_{n-1}\}$.
Mediatoarea segmentului $[P_iP_j]$ împarte planul în două semiplane.
Notăm cu $H(P_i, P_j)$ semiplanul care include punctele mai aproape de P_i decât de P_j.

Definiție

Poligonul Voronoi al lui P_i este $V(i) = \bigcap_{i \neq j} H(P_i, P_j)$, adică intersecția semiplanelor punctelor mai apropiate de P_i decât orice alt punct P_j.
Partiția planului în cele n regiuni $V(0), \ldots, V(n-1)$ se numește diagrama Voronoi asociată lui S și o notăm cu $Vor(S)$.

Așadar, LOCUL GEOMETRIC AL CELOR MAI APROPIATE PUNCTE cere construcția diagramei Voronoi.
c)
Pentru un P_i dat, cel mai apropiat vecin se gaseste intr-o regiune vecina din diagrama Voronoi. Urmatorul algoritm rezolva CEI MAI APROPIATI VECINI:
1. se construieste diagrama Voronoi, reprezentata cu PSLG.
2. se parcurge structura PSLG descriind doagrama Voronoi si pentru fiecare regiune definita de un Pi sinvestigheaza regiunile vecine pentru a se determina cel mai apropiat vecin.

Preprocesare nu exista iar postprocesarea (parcurgerea PSLGului) necesita timpul O(n).

d)

Teoremă

a) Dacă P are complexitatea timp $\Omega(f(n))$ și $P \propto_{g(n)} Q$ atunci Q are complexitatea timp $\Omega(f(n) - g(n))$.

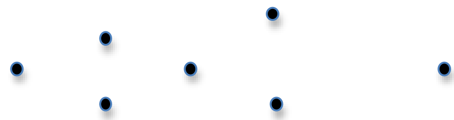
b) Dacă Q are complexitatea $O(f(n))$ și $P \propto_{g(n)} Q$ atunci P are complexitatea $O(f(n) + g(n))$.

Se stie c LOCUL GEOMETRIC AL CELOR MAI APROPIATE PUNCTE are complexitatea O(n log n) (algoritmul facut la curs). Se ia P = CEI MAI APROPIATI VECINI si Q = LOCUL GEOMETRIC AL CELOR MAI APROPIATE PUNCTE.

e) Se stie ca LOCUL GEOMETRIC AL CELOR MAI APROPIATE PUNCTE are complexitatea Ω(n log n) si se aplica din nou teorema de mai sus.

Observatii.
1) peste 90% nu au stiut raspunsurile la a) si b), cee ce arata ca nu s-au citit cursurile.
2) Enuntarea teoremei ar fi adus jumatate din punctaj daca raspunsul nu ar fi fost corect.
3) Enuntarea altor probleme in locul celor cerute nu a fost punctata.

3. Acest exercițiu se referă la algoritmul divide-et-impera de construcție a diagramei Voronoi pentru mulțimea S din figura alăturată.



a) [0.5p] Să se explice cum se calculează dreapta verticală care divizează mulțimea S în S1 și S2.

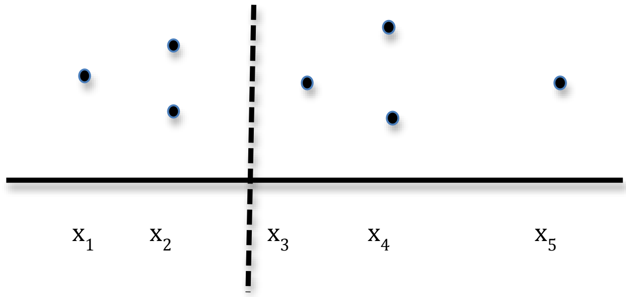
b) [1p] Să se construiască diagramele Voronoi pentru S1 și S2.

c) [1p] Să se arate cum se construiește linia poligonală σ(S1, S2) care separă cele doua diagrame.

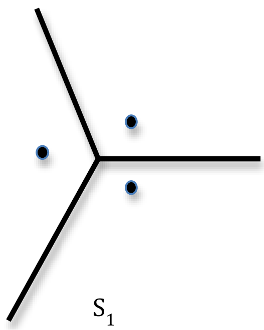
d) [0.5p] Să se arate cum se construiește diagrama Voronoi a lui S din cele douaă diagrame calculate recursiv și σ(S1, S2).

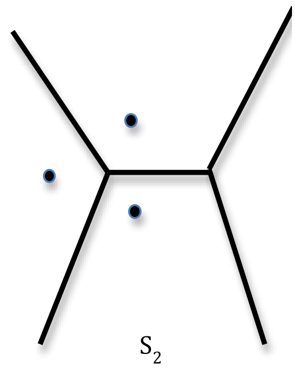
Răspuns.

a) Se calculeaza mediana absciselor. Mediana este definta ca fiind cel de-al n/2-lea element cel mai mic (echivalent, al n/2-lea cel mai mare). Pentru multimea data este x_3. Se duce o verticala la stanga sau la dreapta lui x3. S1 va fi multimea punctelor din stanga vertivalei, S2 multimea punctelor din dreapta.



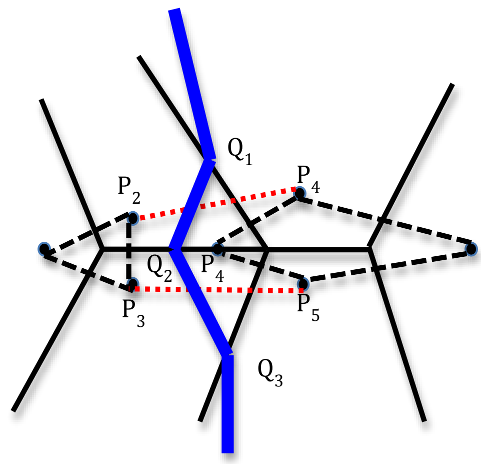
b) Fiind multimi cu putine puncte, se pot construi direct. Se duc mediatoarele segementelor care unesc punctele. Orice punct Voronoi va fi intersectia mediatoarelor unui triunghi. Pentru multimile considerate avem numai regiuni nemarginite.



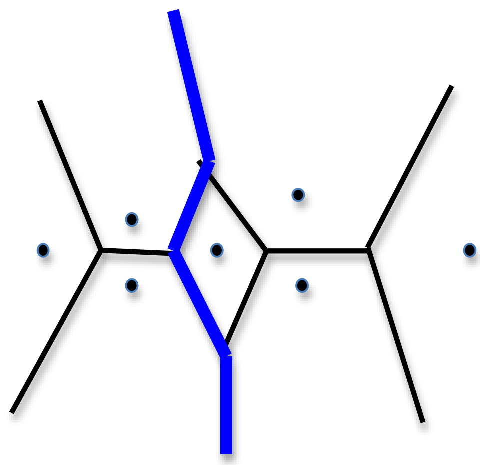


c)

1. Se construiesc infasuratorile convexe, unind punctele din regiunile nemarginite.
2. Se determina dreptele suport P_2P_4 si P_3P_5 pentru cele doua infasuratori.
3. Se duce perpendiculara pe dreapata suport P_2P_4 superioara pana intalneste prima muchie Voronoi in Q_1.
4. Se continua cu perpendiculara pe P_2P_4 pana intalneste cea de-a doua linie Voronoi in Q2.
5. Se continua cu perpendiculara pe P_3P_4 pana intalneste cea de-a doua linie Voronoi in Q3. Aici se intalnest cu perpendiculara pe ce-a de-a doua dreapta suport P_3P_5 si procesul se termina.



d) Se fac intersectiile dintre diagramele Voronoi pentru S1 si S2 si linia poligonala determinata la c). Se sterge din Vor(S2) ce e la stanga liniei si din Vor(S1) ce e al dreapta liniei.



Observatii.

1) Desenele fara explicatii nu s-au punctat deloc sau s-au punctat partial acolo unde din context reiesea cat de cat ideea din spatele constructiei.

2) Multi dintre cei care au scris cate ceva la acest exercitiu, care descrie solutia problemei 2b), nu au stiut sa formuleze problema 2b). Se poate sa se stie solutia dar nu si problema pe care o rezolva?