Solució de l'Examen Parcial EDA 23/03/2015

Proposta de solució al problema 1

a) $\lfloor n/2 \rfloor - 1$

b) 0

c) $\lfloor n/17 \rfloor - 1$

d) El cost de l'algorisme es pot expressar com:

$$\sum_{\substack{x=2 \\ x \text{ primer}}}^{n} \Theta(\lfloor n/x \rfloor - 1) \quad + \sum_{\substack{x=2 \\ x \text{ no primer}}}^{n} \Theta(1).$$

El segon sumatori és $O(n)$. Per altra banda, com que $\lfloor n/x \rfloor - 1$ és $\Theta(n/x)$, el primer sumatori és equivalent a

$$\sum_{\substack{x=2 \\ x \text{ primer}}}^{n} \Theta(n/x) = n \sum_{\substack{x=2 \\ x \text{ primer}}}^{n} \Theta(1/x) = \Theta(n \log\log n).$$

El resultat és doncs

$$\Theta(n \log\log n) + O(n) = \Theta(n \log\log n).$$

e) El cost no millora, continua essent $\Theta(n \log\log n)$, perquè el cost en aquest cas té una expressió similar a l'anterior amb l'única diferència que ara els sumatoris arriben només fins a $\sqrt{n}$. Per tant, l'expressió final que un obté és $\Theta(n \log\log \sqrt{n})$, que és el mateix que $\Theta(n \log\log n)$, ja que $\log\log\sqrt{n} = \log(\frac{1}{2}\log n) = \log\frac{1}{2} + \log\log n = \Theta(\log\log n)$.

Proposta de solució al problema 2

a) El mínim n tal que $n^3 \geq 10n^{2.81}$, és a dir, $n^{0.19} \geq 10$, és $n = \lceil 10^{\frac{1}{0.19}} \rceil = 183299$.

b) El mínim n tal que $10n^{2.81} \geq 100n^{2.38}$, és a dir, $n^{0.43} \geq 10$, és $n = \lceil 10^{\frac{1}{0.43}} \rceil = 212$.

Proposta de solució al problema 3

a) Una solució consisteix en ordenar els intervals en ordre creixent per l'extrem esquerre en temps $\Theta(n \log n)$, i després processar-los de la manera descrita a continuació. Recorrem la seqüència d'esquerra a dreta mantenint l'extrem esquerre mínim (*eem*) i l'extrem dret màxim (*edM*) vistos des de l'últim interval que hem escrit a la sortida. Si el següent interval de la seqüència te un extrem esquerre més gran que l'*edM* llavors podem *tancar* l'interval [*eem*,*edM*], afegir-lo a la sortida, i actualitzar *eem* i *edM* als extrems esquerre i dret de l'interval que acabem de processar. En cas contrari actualitzem l'*edM* si és necessari, és a dir, si l'extrem dret de l'interval processat és més gran que l'*edM*. El cost d'aquesta fase és $\Theta(n)$ i per tant el cost total és $\Theta(n \log n)$.[1]

[1] Una segona solució seria un algorisme de dividir-i-vèncer semblant a l'ordenació per fusió.

b) En primer lloc calculem la unió dels intervals igual que en el primer apartat, amb cost $\Theta(n \log n)$. Després determinem, per cada punt p_i, si està dins d'algun interval de la unió o no. Quan m és gran, com és el cas si $m = n$, una bona solució consisteix a ordenar p en ordre creixent, i després "fusionar" intervals i punts en temps lineal. A cada pas de la fusió considerem un interval $[a_i, b_i]$ i un punt p_j. Si $b_i < p_j$, l'interval es descarta i avancem a la seqüència d'intervals. Si $a_i \leq p_j \leq b_j$, incrementem el comptador i avancem a les dues seqüències. Si $p_j < a_i$, el punt es descarta i avancem a la seqüència de punts. Quan no quedin punts, el procés acaba. El cost de la segona fase és $\Theta(m \log m) + \Theta(n + m)$. Per $m = n$, això és $\Theta(n \log n)$.

c) Farem servir un altre algorisme. Un punt pertany a la unió si i només si pertany a algun dels intervals de la seqüència d'entrada, i per tant podem determinar si pertany a la unió en temps $\Theta(n)$ simplement recorrent la seqüència d'intervals tal com ens ve donada (sense processar-la prèviament). Donat que $m \leq 5$, això són no més de 5 recorreguts de cost $\Theta(n)$ cadascún i per tant el cost total és $\Theta(n)$.[2]

Proposta de solució al problema 4

a) Resposta: $\Theta(3^{\log_2(n)}) \neq \Theta(3^{\log_4(n)})$. Justificació: Siguin $f(n) = 3^{\log_2(n)}$ i $g(n) = 3^{\log_4(n)} = 3^{\log_2(n)/2}$ de manera que $f(n)/g(n) = 3^{\log_2(n) - \log_2(n)/2} = 3^{\log_2(n)/2}$. Com que $\lim_{n \to \inf} f(n)/g(n) = \infty$, $f(n)$ creix estrictament més ràpid que $g(n)$.

b) Cal calcular $2^1 \cdot \cdots \cdot 2^{100} = 2^{5050} \pmod 9$. Com que $2^6 = 1 \pmod 9$ i $5050 = 4 \pmod 6$, tenim $2^{5050} = 2^4 = 7 \pmod 9$.

c) Ordenades d'ordre de creixement més petit a més gran, les funcions són

$$(\ln(n))^2, n^{1/3}, \sqrt{n}, n^4 - 3n^3 + 1.$$

d) Les tres recurrències són:

$A(n) = \Theta(n) + 5A(n/2) = \Theta(n^{\log_2 5})$.
$B(n) = \Theta(1) + 2B(n-1) = \Theta(2^n)$.
$C(n) = \Theta(n^2) + 9C(n/3) = \Theta(n^2 \log n)$.

C és la més eficient perquè $\log n$ creix més lentament que $n^{\log_2 5 - 2} = n^{0.3219\ldots}$.

[2] Una segona solució, menys eficient, seria primer calcular la unió en forma d'intervals disjunts ordenats com en el primer apartat en temps $\Theta(n \log n)$, i després fer m cerques dicotòmiques en temps $\Theta(m \log n)$. Quan m és una constant, el cost total és $\Theta(n \log n)$.