Solució de l'Examen Parcial EDA — 03/11/2022

Proposta de solució al problema 1

(a) El programa anterior escriu el valor N^N per pantalla.

Només cal demostrar que $f(x, n)$ retorna x^n per $n \geq 1$. Es pot demostrar fàcilment per inducció. Per $n = 1$, és obvi que retorna $x = x^1$ gràcies a la primera línia de la funció. Per $n > 1$, si assumim per hipòtesi d'inducció que $f(x, n-1) = x^{n-1}$, aleshores la variable tmp conté x^{n-1}. El bucle suma x vegades el valor de tmp, és a dir x^{n-1}, i el guarda dins *res*. Per tant *res* conté $x \cdot x^{n-1} = x^n$.

Pel que fa al cost, ens n'adonem que la part recursiva de la funció fa tot d'operacions constants més (1) una crida recursiva de mida $n-1$, i (2) un bucle de cost $\Theta(x)$. Seria incorrecte dir que el cost del bucle és $\Theta(n)$ perquè en les successives crides a la funció no es compleix que $x = n$, sinó que x es manté constant i n va decreixent.

Per a solucionar-ho, calcularem d'una banda el cost de la funció sense considerar el bucle i d'una altra banda el cost de totes les execucions del bucle. Si no considerem el bucle, el cost de la funció ve donat per $C(n) = C(n-1) + \Theta(1)$, que té solució $C(n) \in \Theta(n)$. Pel que fa al bucle, tenim en compte que per una crida inicial $f(N, N)$ amb $N > 1$ es faran $N - 1$ execucions del bucle, cadascuna d'elles amb cost $\Theta(N)$. Tot plegat, el bucle ens dona un cost $\Theta((N-1)N) = \Theta(N^2)$. Per tant, el cost del main és $\Theta(N) + \Theta(N^2) = \Theta(N^2)$.

(b) Calculem el límit:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\ln(\ln(n^2))}{\ln(\ln n)} = \lim_{n\to\infty} \frac{\ln(2\ln n)}{\ln(\ln n)} = \lim_{n\to\infty} \frac{\ln(2) + \ln(\ln n)}{\ln(\ln n)} =$$

$$\lim_{n\to\infty} \frac{\ln(2)}{\ln(\ln n)} + \lim_{n\to\infty} \frac{\ln(\ln n)}{\ln(\ln n)} = 0 + 1 = 1$$

Per tant, podem afirmar que $f(n) \in \Theta(g(n))$.

Pel que fa a $F(n)$ i $G(n)$, podríem calcular també $\lim_{n\to\infty} \frac{F(n)}{G(n)}$. Com que quan calculem el límit quan $n \to \infty$ no ens importen els valors que F o G puguin prendre per $n \leq 10$, el límit també serà 1 i per tant $F(n) \in \Theta(G(n))$.

Proposta de solució al problema 2

(a) Recordem que $n = h.size() = s.size()$. La funció proporcionada només fa operacions amb cost $\Theta(1)$ (operacions aritmètiques bàsiques, accessos a vectors, càlcul de mínim i màxim) però té dos bucles ennierats. El bucle intern té cost $\Theta(n)$, i aquest cos és independent de la iteració del bucle extern on estiguem. El bucle extern s'executa n vegades, i per tant el cost total és $\Theta(n \cdot n) = \Theta(n^2)$.

(b) El codi omplert és:

```
int radium_v2 (const vector<int>& h, const vector<int>& s) {
  int r = 0, j = 0;
  for (int i = 0; i < s.size(); ++i){
    while (j < h.size() and h[j] < s[i]) ++j;
    int rad;
    if (j == h.size())  rad = s[i] − h[h.size()−1];
    else if (j == 0)    rad = h[0] − s[i];
    else                rad = min(s[i] − h[j − 1], h[j] − s[i]);
    r = max(r,rad);
  }
  return r;
}
```

Pel que fa al cost, ens fixem que la funció fa tot d'operacions de cost $\Theta(1)$, però té dos bucles ennierats. El que hem de tenir en compte és que el valor de j, que controla el bucle intern, no s'inicialitza en cada volta del bucle extern, sinó que j val 0 a l'inici de la funció i es va incrementant al llarg de tota l'execució, fins a valer com a molt n. Per tant, durant tota l'execució de la funció es fan com a molt n voltes al bucle **while** i, per tant, té cost $\Theta(n)$. Remarquem que aquest cost no és per a cada volta del bucle extern, sinó que ja considera totes les voltes. Finalment només ens queda analitzar el bucle extern, que s'executa n vegades i, per tant, té cost $\Theta(n)$ si obviem el cost del **while**, que ja l'hem comptat a part. Així doncs, el cost total és $\Theta(n+n) = \Theta(n)$.

(c) Una possible solució és:

```
int find (const vector<int>& h, int l, int r, int p) {
  if (l > r) return h.size();
  else {
    int m = (l+r)/2;
    if (h[m] < p) return find(h,m+1,r,p);
    else if (m > l and h[m−1] ≥ p) return find(h,l,m−1,p);
    else return m;
  }
}
```

Per analitzar el seu cost ens n'adonem que totes les operacions tenen cost $\Theta(1)$, excepte la crida recursiva que sempre té mida la meitat de la mida original. Per tant, el cost de la funció ve donat per la recurrència $C(n) = C(n/2) + \Theta(1)$, que té solució $C(n) = \Theta(\log n)$.

(d) La línia que acabem d'introduir sempre té cost en cas pitjor $\Theta(\log n)$. Per tant, com que el bucle **for** dona n voltes, i a cada volta fa un treball $\Theta(\log n)$ més altres operacions de temps $\Theta(1)$, tenim que el cost total és $\Theta(n \log n)$.