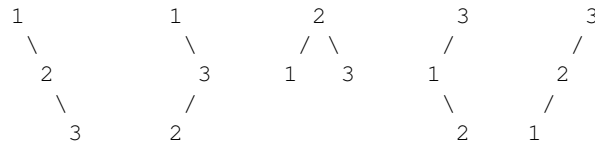


Aquest exercici explora l'ús de vectors per resoldre problemes de programació dinàmica.

1. Oh, no... Un altre cop! Feu una funció `fib :: Int → Integer` que, donat un $n \geq 0$, retorni l' n -èsim nombre de Fibonacci.
2. Feu una funció `binomial :: Int → Int → Integer` que, donat un enter $n \geq 0$ i un enter $0 \leq k \leq n$, retorni el coeficient binomial $\binom{n}{k}$, és a dir, el nombre de formes en què es poden escollir k objectes d'entre un conjunt de n sense tenir en compte l'ordre.
3. Feu una funció `bst :: Int → Integer` que, donat un $n \geq 0$, retorni el nombre d'arbres binaris de cerca amb nodes $1, \dots, n$.

Per exemple, `bst 3` és 5, perquè hi ha 5 arbres binaris de cerca amb nodes 1,2,3:



4. Feu una funció `coins :: [Int] → Int → Int` que, donada una llista de n valors de monedes v_1, \dots, v_n i donat un valor s , trobi el mínim nombre de monedes que sumen s . Cada moneda es pot fer servir diversos (o cap) cops, $s \geq 0$ i $v_i > 0$ per a tot i .
5. Donades dues matrius amb dimensions $n_1 \times n_2$ i $n_2 \times n_3$, el cost de l'algorisme habitual per multiplicar-les és $\Theta(n_1 n_2 n_3)$. Per senzillesa, considerem que el cost és exactament $n_1 n_2 n_3$.

Suposem que hem de calcular $M_1 \times M_2 \times \dots \times M_m$, on cadascuna de les M_i és una matriu amb dimensions $n_i \times n_{i+1}$. Com que el producte de matrius és associatiu, es pot triar en quin ordre es fan les multiplicacions. Per exemple, per calcular $M_1 \times M_2 \times M_3 \times M_4$, es podria fer $(M_1 \times M_2) \times (M_3 \times M_4)$, amb cost $n_1 n_2 n_3 + n_3 n_4 n_5 + n_1 n_3 n_5$, o bé $M_1 \times ((M_2 \times M_3) \times M_4)$, amb cost $n_2 n_3 n_4 + n_2 n_4 n_5 + n_1 n_2 n_5$, o bé tres altres ordres possibles.

Feu una funció `mult :: [Int] → Int` que trobi el cost mínim de calcular $M_1 \times M_2 \times \dots \times M_m$, donades les dimensions $n_1, n_2, \dots, n_m, n_{m+1}$.

Puntuació

Per a cada apartat, hi ha dos tipus de jocs de proves segons la talla de la seva entrada: els petits i els grans. Els petits es poden resoldre recursivament i donen 5 punts cadascún. Els grans requereixen programació dinàmica i donen 15 punts cadascún.

Exemple d'entrada

```
map fib [0..6]
map (binomial 6) [0..6]
map bst [0..6]
coins [1,3,5] 11
```

```
coins [4, 6] 11
mult [10, 20, 30, 40]
mult [9000, 4000, 3500, 8000, 2000, 7500, 6000, 1000, 8500, 5500, 7000]
```

Exemple de sortida

```
[0, 1, 1, 2, 3, 5, 8]
[1, 6, 15, 20, 15, 6, 1]
[1, 1, 2, 5, 14, 42, 132]
Just 3
Nothing
18000
302250000000
```

Informació del problema

Autor : Jordi Petit i Salvador Roura
Generació : 2024-04-30 18:07:51

© *Jutge.org*, 2006–2024.
<https://jutge.org>