

Wstęp do programowania, potok funkcyjny

III kolokwium

2013/2014

Uwaga: każde zadanie proszę rozwiązać na osobnej, podpisanej kartce.

1. Dany jest graf nieskierowany, którego wierzchołki są ponumerowane od 0 do $n - 1$. Napisz procedurę `path : unit → int`, która wyznacza długość (liczoną liczbą krawędzi) najdłuższej ścieżki w tym grafie, na której numery wierzchołków tworzą ciąg rosnący.

Dany graf jest już stworzony i dostępne są następujące operacje na nim:

```
type graph
val size : graph -> int
val neighbours : graph -> int -> int list
val insert_edge : graph -> int -> int -> unit
```

2. W korytarzu harczą myszy. Korytarz ma długość n metrów. Dana jest tablica n nieujemnych liczb całkowitych a opisująca gdzie jest ile myszy, dokładnie na odcinku od i do $i + 1$ metrów od wejścia do korytarza jest $a.(i)$ myszy.

Dysponujesz k kotami. Twoim zadaniem jest takie rozmieszczenie kotów w korytarzu, żeby złapały jak najwięcej myszy. Każdy kot może pilnować ustalonego przez Ciebie spójnego fragmentu korytarza (na przykład, może mieć założoną smycz odpowiedniej długości, przymocowaną do podłogi pośrodku pilnowanego fragmentu korytarza). Fragmenty korytarza pilnowane przez różne koty nie mogą zachodzić na siebie (żeby koty się nie pobiły a smycze nie poplątały), choć mogą się stykać. Niektóre fragmenty korytarza mogą pozostać niepilnowane przez żadnego kota.

Kot, który pilnuje fragmentu korytarza od i do j metrów od wejścia (dla $i < j$), na którym znajduje się $s = a.(i) + a.(i + 1) + \dots + a.(j - 1)$ myszy, złapie: $\max(s - (j - i - 1)^2, 0)$ myszy.

Na przykład, dla $k = 2$ oraz $a = [1; 5; 1; 4; 3; 2; 7; 0]$ poprawnym wynikiem jest 14, jeden kot może pilnować fragmentu korytarza od 1 do 4 metrów od wejścia (łapiąc 6 z 10 myszy), a drugi może pilnować fragmentu korytarza od 4 do 7 metrów od wejścia (łapiąc 8 z 12 myszy).

Napisz procedurę `myszy : int → int array → int`, która dla danej liczby $k \geq 0$ oraz tablicy a , wyznaczy maksymalną liczbę myszy, jakie złapią koty.