Problema pqstr – descrierea soluției

autor: prof. Ionel-Vasile Pit-Rada, Colegiul National Traian, Drobeta Turnu Severin

Soluție O(N*(Q-P)) (20 puncte)
Introducem următoarea definiție:
best[i] = cea mai mare valoare a expresiei ce se poate obține printr-un subșir ce se termină în elementul de la poziția i

Evident un subșir ce se termină la poziția i, poate fi conectat de un element j ce satisface
i-q <= j <= i-p și j >= 0
Astfel deducem recurența:
best[i] = max{ best[j] + abs(A[i] - A[j]) | i-q <= j <= i-p && j >= 0 }

Recurența de față se poate rezolva ușor cu o complexitate O(N*(Q-P)).

Soluție O(N*P) (45 puncte)
Dacă în soluția anterioară se observă că pentru Q>2*P putem înlocui Q cu 2*P, atunci se obține complexitatea O(N*P) cu care se pot obține 45 puncte.

Soluție O(N) (100 puncte) - autor: Mihail-Cosmin Pit-Rada
În contextul observațiilor de mai sus, observăm că prezența modulului abs(A[i] - A[j]) complică simplificarea recurenței.

(1) Să presupunem că modulul nu ar exista, iar recurența ar arata astfel:
best[i] = max{ best[j] + A[i] - A[j] | i-q <= j <= i-p && j >= 0 }

Întrucât A[i] este independent de j, putem observa că:
best[i] = A[i] + max{ best[j] - A[j] | i-q <= j <= i-p && j >= 0 }
echivalent cu
best[i] - A[i] = max{ best[j] - A[j] | i-q <= j <= i-p && j >= 0 }
echivalent cu
minus[i] = max{ minus[j] | i-q <= j <= i-p && j >= 0 }, unde am notat minus[k] = best[k] - A[k]

Practic minus[i] este maximul dintr-un interval [i-q, i-p] al vectorului minus[] (elemente calculate anterior). Avem de a face cu o fereastra glisanta, "sliding window", de dimensiune constanta, q-p+1, ce avansează spre dreapta pe măsură ce i-ul creste. Această problemă se poate rezolva în timp O(N) cu o structura de date de tip "deque", adică o coadă dublă.

(2) Cum putem reduce problema originala la cazul de mai sus?

Observăm matematic că:
best[j] + abs(A[i] - A[j]) = max(best[j]-A[j] + A[i], best[j] + A[j]-A[i])=max(minus[j] + A[i], plus[j] - A[i])
unde minus[k] = best[k] - A[k] si plus[k] = best[k] + A[k]

Pe baza observației anterioare putem calcula doua rezultate:
res_minus = max{ minus[j] + A[i] | i-q <= j <= i-p && j >= 0 }
res_plus = max{ plus[j] - A[i] | i-q <= j <= i-p && j >= 0 }
iar best[i] va fi maximul dintre cele două.

Evident având best[i], putem deduce minus[i] si plus[i] asigurând progresul recurentelor.
Astfel având calculate tripletele (best[0], minus[0], plus[0]), (best[1], minus[1], plus[1]) ... (best[i-1], minus[i-1], plus[i-1])
putem calcula (best[i], minus[i], plus[i]), ș.a.m.d

Cele doua rezultate sunt de tipul problemei (1).

Poate părea paradoxal faptul ca problema a fost simplificata prin adăugarea mai multor valori la recurenta inițială. Cu toate acestea, problema a fost redusa la doua probleme mai simple, iar prin combinarea celor doua soluții se obține soluția problemei inițiale.