Oraşe şi centrale atomice

Metoda ungară

Descrierea problemei

În ţara Letiţia există `N` oraşe şi `M` centrale atomice. Se cunosc poziţiile pe hartă (în coordonate `x, y`) ale tuturor oraşelor şi ale tuturor centralelor atomice. Se cere găsirea soluţiei de cost minim în care fiecare oraş este conectat la o centrală atomică. La o centrală atomică nu se pot conecta mai multe oraşe. Costul unei astfel de conexiuni este egal cu distanţa Manhattan dintre oraş şi centrala atomică.

Formalizând problema, avem un graf bipartit cu două mulţimi de noduri. Se cere rezolvarea problemei cuplajului maxim de cost minim folosind metoda ungară (algoritmul lui Kuhn).

Pentru nota 10 se va considera doar cazul `N = M`, deci fiecare oraş va fi conectat la o centrală, iar la fiecare centrală va fi conectat un oraş. Pentru bonus se va considera şi cazul `M > N` în care vor exista centrale la care nu se va conecta niciun oraş.

Algoritmul lui Kuhn (metoda ungară)

Mai jos se descriu paşii metodei ungare pentru cazul `N = M`. Pentru bonus se va indica o extindere simplă a algoritmului.

1. Se construieşte o matrice `A` de dimensiune `M * N` ce conţine pe poziţia `(i, j)` costul conectării oraşului `j` la centrala `i`. Liniile vor corespunde, deci, centralelor, iar coloanele vor corespunde oraşelor.
2. Se scade de pe fiecare linie minimul ei valoarea minimă de pe acea linie.
3. Se scade de pe fiecare coloană valoarea minimă de pe acea coloană.
4. Se calculează numărul minim de linii şi coloane necesare pentru a acoperi toate valorile zero. Dacă acest număr este `N`, se sare la pasul 6. Dacă acest număr este mai mic decât N, se continuă cu pasul 5.
5. Se identifică cea mai mică valoare neacoperită de nicio linie. Se scade acea valoare din liniile neacoperite şi se adaugă tuturor coloanelor acoperite.
6. Se extrage soluţia alegând `N` valori zero din matrice (exact una pe fiecare linie şi pe fiecare coloană). Poziţiilor acelor valori corespund cuplajului din graf care dă soluţia optimă.

Pentru cazul în care `M > N` (există mai multe centrale decât oraşe), se adaugă `M - N` oraşe fictive care se află la distanţă zero de toate centralele. Practic matricea `A` de mai sus devine

pătratică prin adăugarea unor coloane suplimentare cu valori zero. Desigur, aceste oraşe fictive nu vor afecta costul soluţiei optime şi vor fi ignorate în cuplajul final.

Datele de intrare şi datele de ieşire

Un fişier de intrare va avea `N + M + 1` linii. Pe prima linie se află numerele `N` şi `M` separate de spaţiu. Pe liniile `2 ... N + 1` se află coordonatele celor `N` oraşe, iar pe liniile `N + 2 ... N + M + 1` se află coordonatele celor `M` centrale electrice. Coordonatele sunt numere întregi separate prin spaţiu. Se vor folosi numere întregi pentru a se putea verifica cu precizie egalitatea cu zero în algoritmul ungar.

Fişierul de ieşire va avea `N` linii cu următoarea semnificaţie: pe linia `j` se va găsi numărul centralei la care se conectează oraşul `j`. Deci pe fiecare dintre cele `N` linii se va găsi un număr întreg între `0` şi `M - 1`.

Arhiva cu tema

Tema va fi însoţită de `T` teste şi un script de verificare. Fişierele de intrare se vor numi `test<i>.in` cu `i` fiind un număr de la `0` la `T - 1`. Asociate vor fi fişierele `correct<i>.out` ce vor conţine soluţia optimă pentru fiecare test. Scriptul `test.sh` va apela executabilul `electricity` cu argumentele `./tests/test<i>.in` şi `test<i>.out` şi va compara `fişierul test<i>.out` cu fişierul `correct<i>.out`. Pentru punctaj maxim trebuie trecute toate testele pentru care `N = M`. Celelalte teste sunt pentru bonus.

Arhiva va conţine fişierele sursă, un fişier README cu explicaţii asupra implementării, precum şi un `Makefile` cu regulile `build` şi `clean`. Regula `build` compilează sursele şi produce executabilul `electricity`. Regula `clean` şterge acest executabil şi orice alte fişiere produse în timpul compilării.

Referinţe

http://www.math.harvard.edu/archive/20_spring_05/handouts/assignment_overheads.pdf
https://en.wikipedia.org/wiki/Hungarian_algorithm#Bigraph_formulation