Problema **curiosity**, descrierea soluției

Autor, prof. **Eugen Nodea** – Colegiul Național "Tudor Vladimirescu" – Târgu Jiu

Pentru început câteva observații:

1) deși robotul se deplasează dintr-un punct atât stânga cât și dreapta, pentru că `Xs<Xf` este suficient sa analizăm doar deplasarea spre dreapta, cazul spre stânga fiind similar.

2) între oricare două puncte `x1`, `x2` energia minim consumată =`|x1-x2|`

3) avem soluție numai dacă distanța între oricare două puncte consecutive este mai mica sau egală cu `C`

4) pentru fiecare punct de alimentare este suficient să analizăm cele mai apropiate puncte față de acesta de tip `1`, tip `2`, tip `3`.

Ideea de rezolvare: pentru fiecare punct `x[i]` maximizăm încărcările de tip `3`, dacă nu se mai poate avansa maximizăm pe cât posibil încărcările de tip `2`, ş.a.

Programare dinamică de tip rucsac:

* `tip` - tip de alimentare `{1,2,3}`

* `j` – indice punct de alimentare (se pleacă de la `Xf` către `Xs`)

* `cost[tip][j] = cost[tip][j+1] + max(distanta(x[j+1]-x[j]) - C, 0)`

Complexitate: `O(n)`

Descrierea solutiei pentru problema **curiosity**

Autor: Mihail-Cosmin Pit-Rada

(0) Putem presupune ca `XS` < `XF`. Daca nu este asa putem oglindi toate punctele. Putem astfel presupune ca deplasarea se face de la stanga spre dreapta.

Ne intereseaza doar punctele intermediare din intervalul [`XS, XF`]. Initial bateria este incarcata la capacitatea maxima si este inutila deplasarea in stanga lui `XS`. Deplasarea catre un punct din dreapta lui `XF` din nou este inutila.

(1) Problema are solutie daca si numai daca distanta dintre doua puncte consecutive este cel mult egala cu `C`. Necesitatea: In cel mai bun caz, la un moment dat, bateria este incarcata la maximum, `C`, deci se pot parcurge cel mult `C` unitati distanta. Suficienta: In fiecare punct se poate alimenta pana la capacitatea maxima `C` si astfel se poate ajunge la orice punct care nu este mai departe de `C` unitati distanta.

(2) Vom incerca sa nu facem nici o alimentare, pe cat se poate. La un moment dat ne vom situa in punctul `X` si dorim sa ne deplasam in punctul `Y`. Ce se intampla daca energia curenta nu este suficienta? Inseamna ca trebuia sa alimentam fie undeva mai devreme, fie va trebui sa

alimentam in punctul curent (eventual si in mai multe puncte). Vom incerca din nou sa fim cat de economic se poate si anume, vom alimenta exact cate unitati sunt necesare pentru a face posibila deplasarea, dupa care rationamentul il vom relua similar din punctul `Y`.

(3) Avem trei tipuri de puncte de alimentare `1`, `2`, `3`. Vom demonstra ca e suficient sa luam in calcul, ca potentiale puncte de alimentare, doar cele mai apropiate/recente puncte, cate unul din fiecare tip.

Sa presupunem ca rationamentul este gresit si va trebui, pentru atingerea optimului, sa alimentam si la un punct mai indepartat.
Fie `A` si `B` doua puncte de alimentare de acelasi tip, `A` fiind cel mai apropiat. Daca trebuie alimentate suplimentar `E > 0` unitati in punctul `B`, practic pe drumul de la `B` la `X`, in fiecare moment bateria va avea cu `E` unitati mai mult. Daca acest lucru este posibil din B, cu siguranta acelasi lucru il putem obtine, alimentand doar in `A`.

(4) Intrucat timpii de incarcare variaza in functie de tip, vom prefera sa facem alimentarea la un punct de tip `3`, fiind cel mai rapid. Cand acestul lucru nu este posibil alimentam la un punct de tipul `2`, etc.

(5) Astfel monitorizam cele mai apropiate `3` puncte de alimentare, de fiecare tip cate unul. Calculam de asemenea si cate unitati de energie mai sunt disponibile in fiecare punct, astfel incat sa ne asiguram ca nu depasim capacitatea maxima `C`. Sa intelegem schimbarile ce intervin.

`available(X)` = numarul de unitati de energie ce se pot fi obtinute din punctul `X`. Mai exact, bateria la momentul vizitarii punctului `X`, avea fix `C` - available(`X`) unitati.

```
            L          P           R
-----------|----------|-----------|------------------->
```

Daca dorim sa ne intoarcem in timp, si sa alimentam suplimentar `E` unitati in punctul P se vor intampla urmatoarele:

a) Se consuma E unitati din cele disponibile in `P`
=> `available(P) -= E`

b) In punctele din dreapta lui `P`, bateria va avea cu `E` unitati mai mult, deci in acele puncte disponibilitatea va scadea cu `E`, pentru a nu depasi capacitatea maxima.
=> `available(R) -= E`

c) Disponibilitatea in punctele din stanga lui `P` se va modifica astfel incat, daca vom alimenta cu disponibilitea maxima, cand vom ajunge in `P` capacitatea sa nu depaseasca `C`.
=> `available(L) = min(available(L), available(P))`

Cu cele descrise mai sus se poate obtine o complexitate `O(N)`.