Postavljanje rutera

Problem: Duž jedne ulice su ravmnomerno raspoređene zgrade (rastojanje između svake dve susedne je jednako). Za svaku zgradu je poznat broj korisnika koje novi dobavljač interneta treba da poveže. Odrediti u koju od zgrada treba postaviti ruter tako da bi ukupna dužina optičkih kablova kojim se svaki od korisnika povezuje sa ruterom bila minimalna (računati samo dužinu kablova od zgrade do zgrade i zanemariti dužine unutar zgrada). Na primer, ako je broj korisnika po zgradama jednak $3, 5, 1, 6, 2, 4$, ruter treba postaviti u četvrtu zgradu sleva i dužina kablova je tada jednaka $3 \cdot 3 + 2 \cdot 5 + 1 \cdot 1 + 1 \cdot 2 + 2 \cdot 4 = 30$.

Naivno rešenje bi podrazumevalo da se izračuna dužina kablova za svaku moguću poziciju rutera i da se odabere najmanji. Da bismo izračunali dužinu kablova, ako je ruter u zgradi na poziciji k, računamo zapravo zbir

$$\sum_{i=0}^{n-1} |k-i| \cdot a_i,$$

gde je a_i broj korisnika u zgradi i. Tu težinsku sumu možemo izračunati u vremenu $O(n)$, pa pošto se ispituje n pozicija, algoritam bi bio složenosti $O(n^2)$.

Mnogo bolje rešenje i linearni algoritam možemo dobiti ako primenimo princip inkrementalnosti. Razmotrimo kako se dužina kablova menja kada se ruter pomera sa zgrade k na zgradu $k+1$. Ako je ruter na zgradi k tada je dužina kablova jednaka

$$d_k = \sum_{i=0}^{k-1} (k-i) \cdot a_i + \sum_{i=k+1}^{n-1} (i-k) \cdot a_i.$$

Ako je ruter na zgradi $k+1$, tada je dužina kablova jednaka

$$d_{k+1} = \sum_{i=0}^{k} (k+1-i) \cdot a_i + \sum_{i=k+2}^{n-1} (i-k-1) \cdot a_i.$$

Razlika između te dve sume jednaka je

\begin{align_} d_{k+1} - d_k &= \left(\sum_{i=0}^{k} (k+1-i) \cdot a_i - \sum_{i=0}^{k-1} (k-i)\cdot a_i\right) + \left(\sum_{i=k+2}^{n-1} (i-k-1)\cdot a_i - \sum_{i=k+1}^{n-1} (i - k) \cdot a_i\right)\ &= \left(\sum_{i=0}^{k-1} ((k+1-i) - (k-i)) \cdot a_i\right) + a_k - a_{k+1} + \left(\sum_{i=k+2}^{n-1}((i-k-1) - (i-k)) \cdot a_i\right)\ &= \sum_{i=0}^{k-1}a_i + a_k - a_{k-1} - \sum_{i=k+2}^{n-1} a_i\ &= \sum_{i=0}^ka_i - \sum_{i=k+1}^{n-1}a_i \end{align_}

Dužinu kablova za ruter u zgradi $k+1$ dobijamo od dužine kablova za ruter u zgradi k tako što tu dužinu uvećamo za ukupan broj stanara zaključno sa zgradom k i umanjimo je za ukupan broj stanara počevši od zgrade $k+1$. To je zapravo intuitivno prilično jasno i bez prethodnog komplikovanog matematičkog izvođenja. Pomeranjem rutera za dužinu jedne zgrade nadesno, svakom stanaru koji živi zaključno do zgrade k dužina kabla se povećala za jedno rastojanje između zgrada, a svim stanarima od zgrade $k+1$ nadesno se ta dužina smanjuje za jedno rastojanje između zgrada. Ovaj primer lepo pokazuje kako se intuicija i formalno matematičko izvođenje prepliću - kada možemo do rešenja doći intuitivnim putem, ono je obično veoma elegantno i jednostavno. Sa druge strane, ako ne vidimo odmah rešenje problema, matematički aparat nam može pomoći da do njega dođemo.

Ukupne brojeve stanara pre i posle date zgrade možemo takođe računati inkrementalno (pri prelasku na narednu zgradu, prvi broj se uvećava, a drugi umanjuje za broj stanara tekuće zgrade).

Dakle, u programu možemo da pamtimo tri stvari: dužinu kablova d_k ako je ruter na poziciji k, ukupan broj stanara pre_k pre zgrade k i ukupan broj stanara $posle_k$ od zgrade k do kraja. Na početku, kada je $k=0$, prvi broj d_0 moramo eksplicitno izračunati kao $\sum_{i=1}^{n-1} i \cdot a_i$ (za to nam je potrebno vreme $O(n)$), drugi broj treba inicijalizovati na nulu $pre_0 = 0$, a treći na ukupan broj svih stanara $posle_k = \sum_{i=0}^{n-1} a_i$ (i za to nam je potrebno vreme $O(n)$). Zatim za svako k od 1 do $n-1$ računamo $pre_k = pre_{k-1} + a_{k-1}, posle_k = posle_{k-1} + a_{k-1}$ i $d_k = d_{k-1} + pre_k + posle_k$.

```
// krećemo od zgrade 0
// ukupna dužina kablova ako je ruter u tekućoj zgradi
long long duzina_kablova = 0;
for (int i = 0; i < n; i++)
  duzina_kablova += stanara[i] * i;
```

```
// broj stanara pre tekuće zgrade
long long stanara_pre = 0;
// broj stanara od tekuće zgrade do kraja
long long stanara_posle = 0;
for (int i = 0; i < n; i++)
  stanar_posle += stanara[i];

// minimalna duzina kablova
long long min_duzina_kablova = duzina_kablova;

// obrađujemo sve zgrade od 1 do n-1
for (int k = 1; k < n; k++) {
   // ažuriramo brojeve stanara
   stanara_pre += stanara[k-1];
   stanara_posle -= stanara[k-1];
   // ažuriramo duz
   duzina_kablova += stanara_pre - stanara_posle;
   if (duzina_kablova < min_duzina_kablova)
      min_duzina_kablova = duzina_kablova;
}

cout << min_duzina_kablova << endl;
```

Ukupno vreme izvršavanja ovog algoritma je linearno (izvršavaju se tri petlje, svaka složenosti $O(n)$). Recimo i da nismo morali održavati broj stanara od zgrade k do kraja, već smo ga mogli svaki put izračunavati kao razliku između ukupnog broja stanara i broja stanara pre zgrade k (to praktično ne bi uticalo na vreme izvršavanja).