Rezolvarea problemei lui Josephus folosind liste circulare

Ciuperceanu Vlad-Mihai
Grupa 151

1 Aspecte ale proiectării codului

Implementarea propusă pentru problema lui Josephus rezolvă o variantă extinsă a problemei, pentru un număr n dat de persoane, un indice ind al persoanei care începe, precum și pentru un număr k dat, având semnificația că se vor elimina din k în k persoanele (în particular, vom folosi valoarea $k = 3$, însă putem răspunde pentru orice valoare). Valorile vor fi modificate în $main$, prin variabilele cu același nume.

Pentru a construi lista circulară, începem prin a scrie clasa $Node$, care are un atribut key, ce reține valoarea din nodul respectiv, precum și doi pointeri $prev$ și $next$, care pointează spre nodul precedent, respectiv spre nodul următor, din cele care vor alcătui lista circulară. Pe lângă funcțiile de set și get, mai avem și funcția $getNextDeleted$, ce primește ca parametru un număr k și care returnează al k-lea nod de după cel curent, adică nodul care urmează să fie eliminat (pentru noi va conta cel de-al treilea).

Pentru lista circulară, vom implementa clasa $CircularList$, ce va avea ca atribute nodul ce reprezintă capătul listei (nodul care este și începutul și sfârșitul listei, având în vedere că aceasta este circulară) - $head$. De asemenea, pentru a optimiza operația de a afla lungimea listei, vom reține și un atribut $size$, pe care îl vom actualiza după operațiile de inserare și de ștergere.

Aceste operații vor fi implementate prin 2 funcții membru, în care vom adăuga un nod la finalul listei, respectiv vom șterge un nod dat ca parametru, luând în calcul toate cazurile pentru fiecare dintre ele. Am păstrat între comentarii și codul pentru inserarea și stergerea unui nod de la o anumită poziție, dar acestea nu au fost necesare pentru problema noastră.

Revenind la rezolvarea problemei date, vom începe prin a creea lista cu cele n noduri cu valorile de la 1 la n, urmând să rezolvăm problema prin intermediul funcției $josephus$, ce primește ca parametri o listă circulară, indicele celui ce începe și numărul k pentru eliminarea persoanelor (vom folosi, în principal, valoarea 3).

Mai întâi vom pleca din capătul listei și o vom parcurge, până când ne aflăm în nodul ind. Apoi, cât timp mai există noduri în listă, obținem nodul care va fi eliminat, având grijă la cazul în care chiar nodul curent trebuie eliminat.

După ce am făcut eliminarea nodului respectiv, ne mutăm în următorul nod disponibil. La final, returnăm informația din nodul rămas, reprezentând nodul care a supraviețuit. Complexitatea algoritmului va fi $O(n^2)$.

2 Experimente

Dând câteva valori, am obținut următoarele rezultate:

- pentru $n = 4$, $ind = 1$ și $k = 3$, nodul rămas este: 2
- pentru $n = 5$, $ind = 1$ și $k = 3$, nodul rămas este: 4
- pentru $n = 4$, $ind = 2$ și $k = 3$, nodul rămas este: 3
- pentru $n = 50$, $ind = 1$ și $k = 3$, nodul rămas este: 38
- pentru $n = 1000$, $ind = 3$ și $k = 3$, nodul rămas este: 980