Maeștri-cofetari, prăjituri și omuleți flămânzi doresc să treacă un râu...

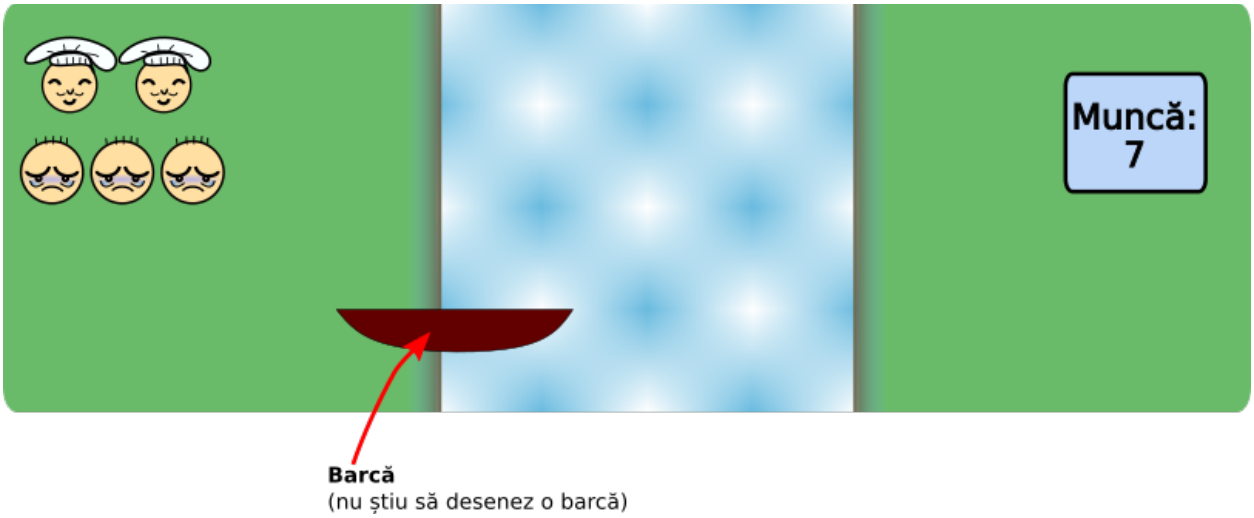
Se va implementa această problemă pornind de la unul dintre exemplele de laborator. Pentru a nu schimba nume de variabile puteți considera un grup de oameni ca fiind canibalii și ceilalți misionarii însă trebuie să specificați cine este fiecare în fișierul **explicatii.txt**.

1) Se vor citi dintr-un fișier de intrare **numerele naturale nenule**:
    a) NC - număr de maeștri-cofetari (îi vom numi pe scurt, cofetari)
    b) NO - număr omuleți flămânzi
    c) C - un cost de bază pentru calcularea costului unei mutări
    d) M - numărul de locuri în barcă
    e) K - cantitatea de prăjituri produse de un singur bucătar în cadrul unui transport cu barca.
    f) W- o cantitate de muncă de efectuat pe malul final.
2) **Inițial** pe malul stâng al râului avem omuleții flămânzi și cofetarii. Pe niciunul dintre maluri nu avem prăjituri, în starea inițială. Aceștia doresc să se deplaseze pe râu cu o barcă având M locuri.
3) **La început, pe malul drept (final) e o cantitate W de muncă** de făcut, care **poate fi realizată doar de omuleții flămânzi**. O unitate de muncă e realizată numai dacă un omuleț flămând a ajuns pe malul final și nu a plecat imediat cu barca (a rămas acolo pâna s-a întors barca măcar încă o dată de când l-a lăsat pe mal). Putem considera că numărul de unități de muncă descrește chiar când ajunge barca cu noii omuleți flămânzi. Un omuleț flămând nu are voie să plece până nu a efectuat unitatea de muncă. Nu va efectua mai multe unități de muncă într-o singură ședere pe mal. Indicație: putem detecta din starea-părinte câți omuleți aveam pe malul final înainte de a venit barca și știm că doar pe aceia îi putem trimite înapoi.. Dacă munca ajunge la 0, omuletii flămânzi pot pleca oricând vor de pe malul finalul.
4) **Scopul problemei** este ca toate cerințele de mai jos să fie îndeplinite în același timp:
    a) toți cofetarii să fie pe **malul final**
    b) toată **cantitatea de muncă de pe malul final** să fie efectuată (**să ajungă egala cu 0**)
    c) omuleții flămânzi să se fi întors toți acasă pe **malul <u>inițial</u>**
    d) **să nu mai existe prăjituri nemâncate pe niciunul dintre maluri**, fiindcă e păcat...
5) Barca are o bucătărie în care cofetarii pot face prajituri. Fiecare cofetar poate face K prăjituri la un transport cu barca de pe un mal pe altul. Prăjiturile sunt depuse pe malul pe care ajunge barca.
6) **Dacă în barcă sunt și cofetari și omuleți flămânzi**, omuleții vor mânca toate prăjiturile, înainte de a ajunge barca la mal, rezultând în faptul că **barca nu va mai depune prăjituri pe malul pe care ajunge**.
7) **Omuleții flămânzi mănâncă prăjituri (de pe malul curent) înainte de a urca în barca și la coborârea din barcă (de pe malul pe care au ajuns).** Dacă nu sunt suficiente prăjituri pe malul pe care ar trebui să mănânce, aceștia refuză să intre sau să iasă din barcă, neefectuându-se transferul. Prin urmare **nu are sens să transportâm omuleți flămânzi dacă nu avem suficiente prăjituri** pe malul celălalt și nici nu-i putem trimite dacă nu avem suficiente pe malul curent.
8) **Barca nu poate să plece fără oameni.** Barca nu mută prăjituri de pe un mal pe altul, doar le depune pe mal pe cele create în transportul curent.
9) **Costul** unui **transport cu barca în care sunt doar cofetari este egal cu *1+C/NR_COFETARI_BARCA*** (C împărțit la câți cofetari sunt în barcă) când sunt doar cofetari în barcă (și deci fac prăjituri) și egal cu ***NR_COFETARI_BARCA*C+COST_FLAMANZI*** **când transportul e mixt** (cofetari alături de omuleți flămânzi. Parametrul *COST_FLAMANZI* este NR_FLAMANZI când barca merge de pe malul inițial spre final si 2*NR_FLAMANZI la întoarcere. NR_FLAMANZI este numărul de omuleți flămânzi din barcă. Dacă **barca mută doar omuleți flămânzi** (numărul de cofetari e nul), **costul întregului transport va fi 1**.

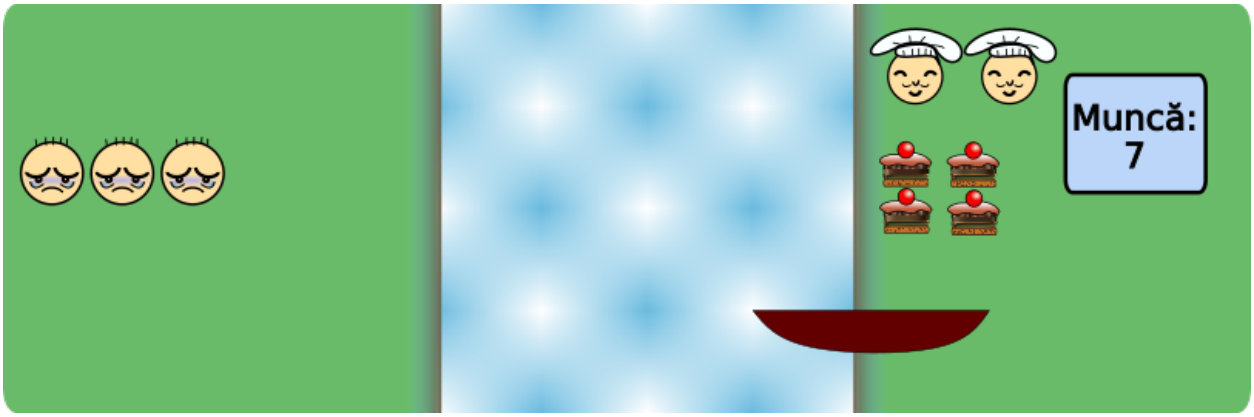
Exemplu de fișier de intrare (se consideră că s-au scris în ordine: NC (cofetari), NO (omuleți flămânzi), C (unitate de cost), M (locuri în barcă), K (număr prăjituri produse), W (cantitatea de muncă))

```
2 3 2 3 2 7
```

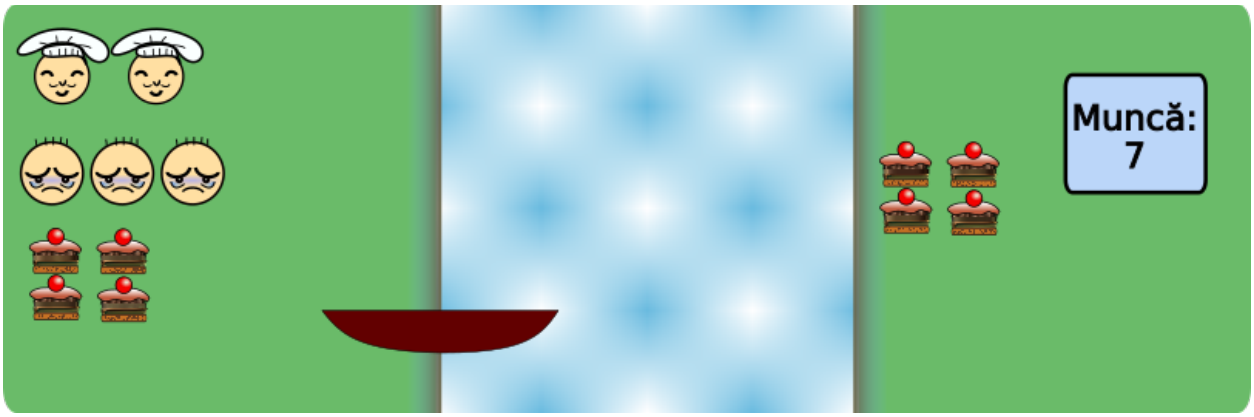
Reprezentare a stării inițiale (cofetarii au bonetă, omuleții flămânzi sunt cei supărați):



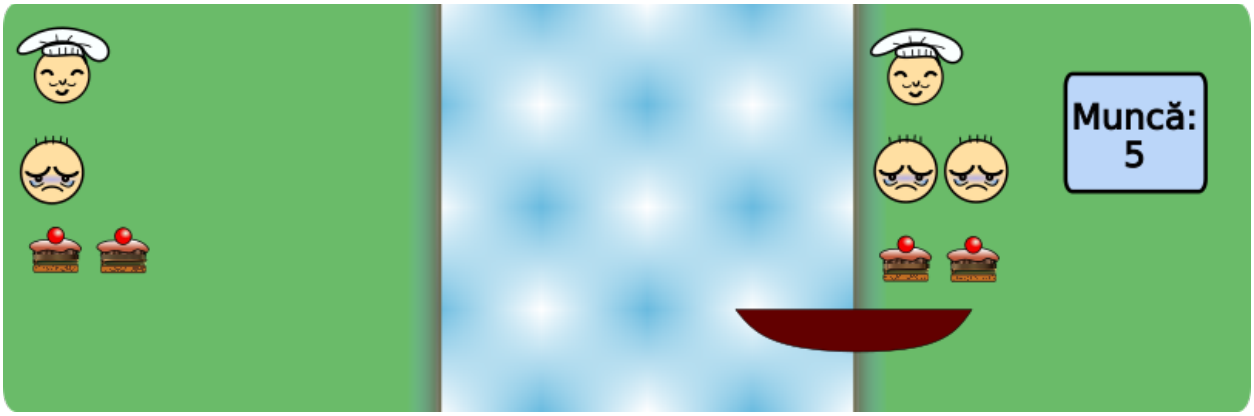
Mai jos e un exemplu de mutare (am mutat 2 cofetari ca să pun prăjituri pe malul final să atrag omuleții flămânzi). Nu puteam să mut și omuleți flămânzi, neavând prăjituri.



Mut înapoi cofetarii, ca să creez prăjituri și pe malul inițial ca să urce omuleți flămânzi în barcă.



Pot deja să mut și omuleți flămânzi, însă îi voi trimite cu un cofetar, deoarece ei nu se pot întoarce de pe malul celălalt până nu au efectuat munca și asta presupune o plecare de barcă de pe malul final care ar fi imposibilă (fiind toți omuleții flămânzi ocupați); deci e necesar un cofetar.



din această stare mai avem doar un succesor, și anume întoarcerea cofetarului.
Nu am mai afișat și restul de mutări până la starea finală.

Cerințe(4.5p+1.5 bonus - punctajul se trunchiaza la 4.5):
Pentru fiecare cerință scrieți un comentariu cu numarul cerinței rezolvate pentru a identifica usor ce ați făcut.
Vom considera o stare ca fiind situația în care se află malurile când barca staționează, iar o tranziție între stari (noduri) va fi o deplasare de barcă.

1. **(0.5p)** Realizați următoarele 3 fișiere de input:
    a. (0.2p din 0.5) Să se scrie un fișier de input fără soluții, numit input_fara_sol.txt. Veți alege valori pentru parametrii problemei astfel încât să fie evident că nu există vreo combinație de mutări care să ducă la rezolvarea problemei. Valorile trebuie să fie însă conform domeniului de valori cerut în enunț (de exemplu un număr natural, dacă parametrul e specificat ca fiind natural etc). Justificați în fisierul explicatii.txt de ce inputul nu are solutii.
    b. (0.2p din 0.5) Să se scrie un fișier de input, input2noduri.txt, care are ca soluție un drum format din 2 noduri (s-a efectuat doar o tranziție) iar în fișierul explicatii.txt scrieți informațiile corespunzătoare celor 2 noduri. În cazul în care un astfel de fișier nu se poate realiza explicați de ce (în explicatii.txt).
    c. (0.1p din 0.5) Să se realizeze un fișier de input, input_initial_final.txt, în care starea inițială este și finală. În cazul în care un astfel de fișier nu se poate realiza explicați de ce (în explicatii.txt).
2. **(0.5p) Modificați citirea din fișier** pentru a salva datele conform formatului cerut pentru fisierul de intrare. Numele malurilor nu se mai citesc, se consideră din start că stâng e initial și drept final. În fișierul explicatii.txt, indicați ce date ar trebui memorate într-o stare și ce structură de date folosim astfel încât să avem toate informațiile necesare, ocupând un minim de memorie pentru un nod. Justificați cum pot fi calculate anumite informații deduse din stare nefiind necesară salvarea în memorie.
3. **(0.25p)** Pentru starea inițială rezultată din fișierul de intrare dat în exemplu, scrieți în fișierul explicatii.txt câți succesori avem. Apoi explicați pe caz general pentru o **stare inițială** rezultată dintr-un fișier de intrare cu restricțiile NC>=3 (cofetari), NO>=3(omuleti flămânzi), C(unitate de cost), M(locuri în barcă), K>=2(număr prăjituri produse), W>=2(cantitatea de muncă)) care este numărul de succesori (o formulă care să cuprindă un subset din parametrii scrișI mai sus: NC,NO,C,M,K,W).
4. **(0.25p)** Modificați funcția de testare a scopului care verifică că s-a ajuns într-o stare finală.
5. **(2p)** Modificați modul de **generare a succesorilor** ca să corespundă cu cerințele de mai sus.
6. **(0.5p) Calculați costul pentru un succesor**. In cazul în care nu faceți subpunctul anterior, puteți rezolva acest subpunct prin crearea unei funcții *cost(nodSuccesor)* care returnează costul în funcție de ce conține acesta și ce ar trebui să conțină nodul său părinte (proprietatea *parinte*).

7. **(1p)** Modificați funcția **calculeaza_h** adaugând o euristică admisibilă în a cărei formulă să se folosească (să aibă rol în expresia formulei de calcul fără a fi anulate, de exemplu înmulțite cu zero) numărul de cofetari dintr-una dintre locații (maluri), numărul de omuleți flămânzi dintr-una dintre locații și valoarea parametrului C. Explicați într-un comentariu scris deasupra antetului funcției ce este o euristică admisibilă și justificați de ce euristica voastră este admisibilă (nu trebuie demonstrație riguroasă) și explicați cum este afectată euristica de costul mutărilor.
8. **(0.25p)** Modificați funcțiile folosite în afișarea soluției astfel încât la fiecare stare afișată să se vadă câți cofetari și omuleți flămânzi avem pe fiecare mal (nu se vor mai afișa "canibali" și "misionari"), câte prăjituri avem pe fiecare mal și care e cantitatea de muncă rămasă pe malul final.
9. **(0.25p)** Modificați algoritmul A* dat astfel încât variabila c(coada) să fie un obiect de tip PriorityQueue (coadă de priorități) din modulul predefinit queue (alternativ va puteti face voi implementarea pentru clasa PriorityQueue după exemplul din exercițiul dat la laborator).
10. **(0.5p)** Afișați pe ecran timpul cumulat pentru calcularea euristicii pe toate apelurile (de exemplu dacă funcția calculeaza_h s-a apelat pe parcursul programului pentru 3 noduri și a durat 1s, 3s respectiv 0.5s, timpul cumulat ar fi 4.5s). Pentru rezolvare puteți folosi fie cProfile si scrieți într-un comentariu în mod clar și detaliat cum identificăm rândul și coloana pe care trebuie să ne uităm în tabelul afisat, fie adăugând cod în program care însumează acești timpi.

Observație: pentru a nu omite trimiterea unui subpunct menționăm că rezolvarea completă a cerințelor presupune predarea următoarelor fișiere (descrise în cerințe):

- **fișierul cu extensia py** cu codul python care rezolvă toate cerințele de implementare (la rularea acestui program pentru un fișier de input dat, trebuie să fie afișate soluțiile în consolă).
- fișierul **input_fara_sol.txt** cu inputul fără soluții (dacă este posibil)
- **input_initial_final.txt** cu starea inițială care e și finală (dacă este posibil)
- **input2noduri.txt** cu inputul care are ca solutie drumul format din doar 2 noduri (o tranzitie) (dacă este posibil)
- fisierul **explicatii.txt** cu eventualele explicații cerute