Laborator 07

Pentru acest laborator nu veți putea compila din Visual Studio, va trebui să compilați din linia de comandă. Aveți un Makefile dat, e suficient pentru primul program să scrieți comanda make în terminal. Citiți Makefile-ul ca să înțelegeți care sunt fișierele din care este construit programul și citiți codul programului.

1. Testați programul **sample-par** și **sample-seq**. Acest program exemplifică cum se poate paraleliza un program recursiv folosind replicated workers. Analizați și fișerele **Workers.h** și **Workers.c**. Va trebui să folosiți N foarte mare pentru paralel.
2. Paralelizați programul **getPath** folosind paradigma replicated workers. getPath-seq caută într-un graf TOATE drumurile non-ciclice de la un nod la altul. Este suficient ca programul paralel să afișeze o singură soluție. Graful default este pus în poza din Figură 1.
3. Scrieți în readme rezultatul sanity check și timpi de execuție. Nu e necesar intensive check. Pentru a testa timpii puteți genera un graf mai mare folosind funcția **generateGraph**. Pentru unele grafuri nu există soluție.
4. Paralelizați programul **colorGraph** folosind paradigma replicated workers. Acest program face colorarea unui graf. Pentru colorarea unui graf se dorește ca două noduri adiacente să aibă culori diferite. Există un număr de culori dat prin variabila COLORS. Fiecare număr, de la 0 la COLORS reprezintă o culoare. Graful default este pus în poza din Figură 1.
    - Mai multe detalii în Colorarea grafurilor.pdf
5. Scrieți în readme rezultatul sanity check și timpi de execuție. Nu e necesar intensive check. Pentru a testa timpi puteți genera un graf mai mare folosind funcția **generateGraph**. Pentru unele grafuri nu există soluție.

În toate cazurile de mai sus este suficient și recomandat să afișați o singură soluție.

Exercițiile de la 1 la 5 sunt **obligatorii**. Conceptele explorate sunt esențiale pentru obținerea notei **minime** de promovare.

Vă recomandăm, pentru a crește șansele de a obține o notă cât mai mare să explorați și următoarele exerciții:

6. Paralelizați programul **queens** folosind paradigma replicated workers.
    - Mai multe detalii în Problema reginelor.pdf
7. Scrieți în readme rezultatul sanity check și timpi de execuție. Nu e necesar intensive check.
8. Implementați un program care să primească două fișiere și verifică dacă liniile unui fișier sunt un subset din liniile altui fișier.
9. Modificați scriptul în așa fel încât să folosească programul scris la punctul anterior în loc de diff pentru a verifica corectitudinea soluțiilor.
10. Implementați o soluție recursivă pentru problema Turnurilor din Hanoi.
11. Paralelizați programul turnurilor din hanoi folosind paradigma replicated workers.

HINTS:

În program graful este reținut ca o listă de adiacență. Fiecare linie a coloanei e o muchie. Muchia **x** conectează nodurile **graph[x][0]** și **graph[x][1]**.

Pentru a converti un program recursiv la paradigma replicated workers trebuie să creați un task inițial, să îl puneți pe coadă. În funcția task-ului (**runTask**) puteți pune vechea funcție recursivă (după ce schimbați parametri pe care aceasta îi primește pentru a se potrivi cu runTask). În loc de apelul recursiv va trebui să creați un nou task și să îl puneți în coada.

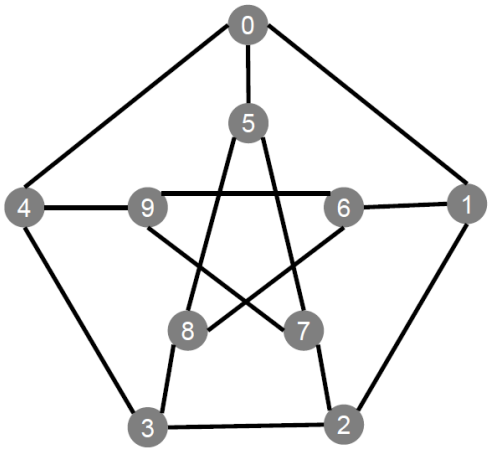
Astfel va trebui să lucrați în main și în funcția recursivă. În main va trebui să porniți thread-urile worker (**startWorkers()**) să adăugați un prim task (**putTask()**) și să așteptați terminarea thread-urilor worker (**joinWorkerThreads()**).

Task-urile trebuie să conțină un set de date dependent de problemă, vă recomandăm crearea propriei structuri pentru a putea adăuga datele de care task-ul are nevoie. Pointerul **data** este dat ca parametru funcției apelate în momentul rulării task-ului.

În momentul creării unui nou task trebuie să vă asigurați că faceți copie completă a celui anterior (inclusiv alocare și copiere de vectori).

Puteți opri programul (**forceShutDownWorkers()**) la găsirea primei soluții sau la găsirea tuturor soluțiilor. Pentru a putea opri în al doilea caz trebuie să știți câte soluții sunt sau de a avea o metodă de a determina care este ultima.

Asigurați-vă că afișarea este thread-safe. Acest lucru poate fi o problemă mai ales la queens.



Figură 1 Graf stea