
PROIECT PROIECTAREA ALGORITMILOR SELECTEREA ACTIVITATIILOR

May 19, 2020

Sotiraq Saraci
Facultate de Automatica, Calculatoare si Tehnologia Informatiei
Calculatoare Romana, Anul I

Contents

1	Introducere si Enuntul problemei	3
2	Algoritmi Propuși	3
3	Datele experimentale	3
4	Proiectarea experimentală a aplicației	5
5	Rezultatele concluziile	5
6	Referințele	5

Abstract

Acest document introduce metodologia folosita de realizare a temelor pentru cursul de Proiectarea algoritmilor. Documentul contine, de asemenea descrierea algoritmilor.

1 INTRODUCERE SI ENUNTUL PROBLEMEI

Problemul se numește Selectare Activităților. Este un problem faimos in campul de Proiectarea Algoritmilor mai specific in Greedy Algorithms. Enuntul problemei e asta: Se consideră o mulțime de activități astfel încât fiecare dintre ele necesită acces exclusiv la o resursă comună. Pentru fiecare activitate se cunoaște timpul de start și durata. Se cere determinarea unei submulțimi de cardinalitate maximă a acestor activități de astfel încât oricare două activități selectate sunt mutual compatibile (nu se suprapun în timp). Trebuie să implementăm doi algoritmi diferiți.

2 ALGORITMI PROPUȘI

ACTIVITY-SELECTION(*start, finish, number, i, j, k*)

1. *read number*
2. **for** *k* = 0 to *number*
3. *read start[k]*
4. *read finish[k]*
5. *read i, j*
6. **if** *start[j] >= finish[i] and start[j] >= finish[i]*
7. **return** 1
8. **else**
10. **return** 0

Figure 1: Algoritm de activitati selectate din utilizator.

3 DATELE EXPERIMENTALE

Structura programului este simpla. Am creat un fisier "function.c" unde sunt sortat algoritmi si un fisier "main.c" unde programul este rulat. Am ales un interval de timp de 12 ori unde ia loc toate activitati. Este cerut din utilizator sa pune numar de activitatii si timpul de inceperea si terminarea. Testul meu contine 5 activitati cu timpuri ca mai jos:

RANDOM-SELECTION(*start*, *finish*, *number*, *i*, *j*, *temp*)

```

1. read number
2. for k = 0 to number
3.   read start[k]
4.   read finish[k]
5. temp = 0
5. while temp not equal to 3
5.   i = generateRandomInt (0, number)
5.   j = generateRandomInt (0, number)
5.   if i = j
5.     temp ++
8.   else temp = 3
6. if start[j] >= finish[i] and start[j] >= finish[i]
7.   return 1
8. else
10. return 0

```

Figure 2: Algoritm de activitati selectate random.

Activity	A[0]	A[1]	A[2]	A[3]	A[4]
Start	2	6	3	1	2
Finish	5	9	6	3	6

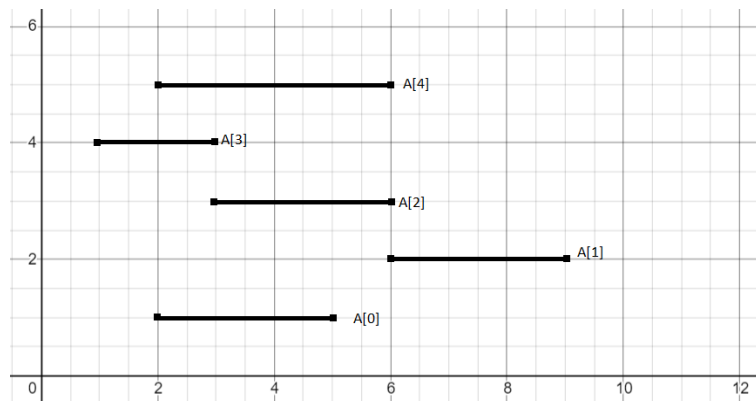


Figure 3: Activitati prezentate intr-un tabel

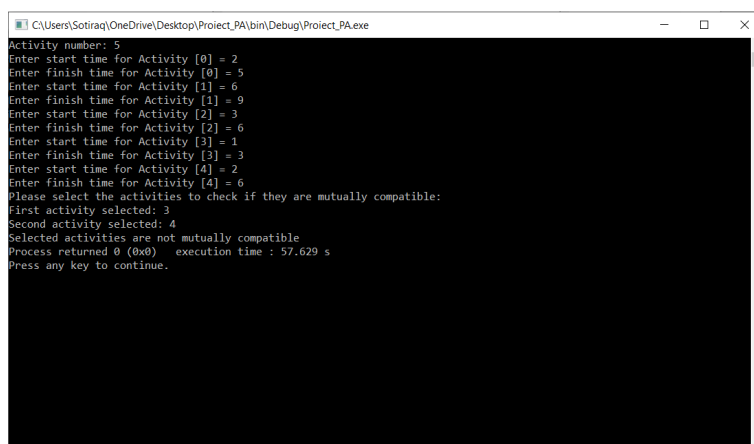
4 PROIECTAREA EXPERIMENTALA A APLICAȚIEI

Dupa ce punem toate date de intrare, algoritmi o sa rulez. In primul algoritm algoritmul o sa compara doua date puse din utilizator. Dupa conditie daca timpul de start de activitatea[i] este mai mica sau egala cu timpul de terminare de activitatea[j] sau timpul de start de activitatea[j] este mai mare sau egala cu timpul de terminare de activitatea[i] algoritmul o sa ruleaza cu succes.

Acesta comparatie este valabil si pentru algoritmul 2, doar ca in algoritmul 2 numarul activitatii este selectat random. De asemenea aici am pus o conditie ca permite executarea algoritmului nu mai putin decat 3 ori in cazul ca activitati selectate random sunt aceeasi. (De exemplu este ales activitate[3] and activitate[3]).

5 REZULTATELE CONCLUZIILE

Dupa ce rulam programul nostru si am pus date de intrare primim rezultatele de mai jos:



```

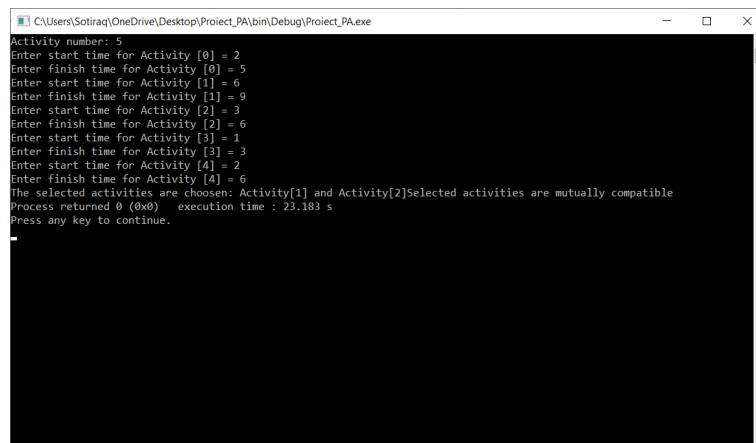
C:\Users\Sotiraq\OneDrive\Desktop\Proiect_PA\bin\Debug\Proiect_PA.exe
Activity number: 5
Enter start time for Activity [0] = 2
Enter finish time for Activity [0] = 5
Enter start time for Activity [1] = 6
Enter finish time for Activity [1] = 9
Enter start time for Activity [2] = 3
Enter finish time for Activity [2] = 6
Enter start time for Activity [3] = 1
Enter finish time for Activity [3] = 3
Enter start time for Activity [4] = 2
Enter finish time for Activity [4] = 6
Please select the activities to check if they are mutually compatible:
First activity selected: 3
Second activity selected: 4
Selected activities are not mutually compatible
Process returned 0 (0x0) execution time : 57.629 s
Press any key to continue.

```

Figure 4: Rularea de algoritmul 1

6 REFERINȚILE

- www.euroinformatica.ro
- www.geeksforgeeks.org
- The C programming Language second edition - Ritchie and Kernighan



```
C:\Users\Sotirag\OneDrive\Desktop\Proiect_PA\bin\Debug\Proiect_PA.exe
Activity number: 5
Enter start time for Activity [0] = 2
Enter finish time for Activity [0] = 5
Enter start time for Activity [1] = 6
Enter finish time for Activity [1] = 9
Enter start time for Activity [2] = 3
Enter finish time for Activity [2] = 6
Enter start time for Activity [3] = 1
Enter finish time for Activity [3] = 3
Enter start time for Activity [4] = 2
Enter finish time for Activity [4] = 6
The selected activities are choosen: Activity[1] and Activity[2]Selected activities are mutually compatible
Process returned 0 (0x0)   execution time : 23.183 s
Press any key to continue.
```

Figure 5: Rularea de algoritmul 2