



1st Junior Balkan Olympiad in Informatics
Belgrade 2007

Day 2 – Task 3

Chairs

First thing is to convert all times in input into number of seconds. Because all entry and exit times are in one day, we get N intervals in range of 0 to $24 * 60 * 60 = 86400$. Problem is to find point that is contain in the most of intervals.

One approach is to make one large array with dimensions $[0..86399]$ and for every interval $[a, b]$ just add 1 to all seconds that are inside this interval (you have to be careful with left and right boundary, and mark them also). This can take at most $86400 * 1000 \sim 87$ millions operations – which is acceptable. Now, we just find the maximum number in array and print the result.

Another way to solve problem is to sort given intervals with respect to left end. We also made one array which represents whether time represents begin or end of current interval – we mark that second with +1 or -1. Now we are iterating through sorted seconds and just add +1 if we run onto begin of some interval, or subtract 1 (or equivalently add -1) if we closed some interval. Of course, we are maintaining some global maximum while going through array. This leads to time complexity $O(N \log N)$, independently of time limits.

Example solution in Pascal:



1st Junior Balkan Olympiad in Informatics Belgrade 2007

Day 2 – Task 3

```
const
  MaxN = 1000;
  DayLenght = 24 * 60 * 60;
var
  n, i : Integer;
  t : LongInt;
  s : String;
  tStart, tEnd : array [1..MaxN] of LongInt;

  a : array[0..DayLenght-1] of Integer;
  sol : Integer;

function HmsToT(h, m, s : Integer) : Longint;
begin
  HmsToT := ((h * 60) + m) * 60 + s;
end;

function ChToN(ch1, ch2 : Char) : Integer;
begin
  ChToN := (Ord(ch1) - Ord('0')) * 10 + (Ord(ch2) - Ord('0'));
end;

begin
  ReadLn (n);
  for i := 1 to n do begin
    ReadLn (s);
    tStart[i] := HmsToT(ChToN(s[1], s[2]), ChToN(s[4], s[5]), ChToN(s[7], s[8]));
    tEnd[i] := HmsToT(ChToN(s[10], s[11]), ChToN(s[13], s[14]), ChToN(s[16], s[17]));
  end;

  for t := 0 to DayLenght - 1 do
    a[t] := 0;
    for i := 1 to n do
      for t := tStart[i] to tEnd[i] do
        inc(a[t]);
      end;
    end;

    sol := 0;
    for t := 0 to DayLenght - 1 do
      if a[t] > sol then
        sol := a[t];
      end;
    end;
  Writeln(sol);
end.
```

Example solution in C#:



1st Junior Balkan Olympiad in Informatics

Belgrade 2007

Day 2 – Task 3

```
using System;
using System.IO;

class Chairs
{
    static void Main(string[] args)
    {
        string s = Console.ReadLine();
        int n = int.Parse(s);
        int [] inTime = new int [n];
        int [] outTime = new int [n];
        for (int i = 0; i < n; i++)
        {
            s = Console.ReadLine();
            string [] p = s.Split(new char [] { ' ', ':' });
            inTime[i] = int.Parse(p [0]) * 60 * 60 +
                        int.Parse(p [1]) * 60 + int.Parse(p [2]);
            outTime[i] = int.Parse(p [3]) * 60 * 60 +
                        int.Parse(p [4]) * 60 + int.Parse(p [5]) + 1;
        }
        int [] time = new int [2 * n];
        int [] mark = new int [2 * n];
        for (int i = 0; i < n; i++)
        {
            time [2 * i] = inTime [i];
            time [2 * i + 1] = outTime [i];
            mark [2 * i] = 1;
            mark [2 * i + 1] = -1;
        }
        Array.Sort(time, mark);
        int sol = 0;
        int sum = 0;
        for (int i = 0; i < 2 * n; i++)
        {
            if ((i > 0) && (time [i] > time [i - 1]) && (sum > sol))
                sol = sum;
            sum = sum + mark [i];
        }
        sol = Math.Max(sol, sum);
        Console.WriteLine(sol);
    }
}
```