

METU – Dept. of Computer Engineering - CENG315

Fall 2012-2013 -- Homework #2

In this programming homework you are going to generate an algorithm for a given problem using different sorting algorithms that will be specified. Here is the problem;

A group of **N** mine workers want to get out of a mine which has a dark and narrow exit path. **Two** people may pass the exit path at any time at most, and each worker group must have a torch to lighten their way. The workers have only **one** torch, so some sort of transport scheme should be arranged in order to return the torch so that all the workers may get out of the mine. Finally, each worker has a specified walking speed and the passing speed of a group is determined by the speed of the slower worker in the group.

Your homework is to generate an algorithm that gets all the workers out of the mine in the **minimum time**.

Specifications of the Homework

- Implement all the following sorting algorithms in your homework. You have to solve the given homework problem using them:
 - 1 Counting Sort
 - 2 Shell Sort (take “gap” as “powers of two minus one”: 1, 3, 7, 15, 31, 63, ...)
 - 3 Quick Sort (take pivot as the first element)
 - 4 A sorting algorithm of your choice (with at most $n \log n$ complexity)
- You need to prepare a written report, including the charts comparing the ***running times of the sorting algorithms*** vs ***N***, and details of your own sorting algorithm.
- You should implement in C/C++.

INPUT Specifications

- Your program is to read from a file named “hw2.inp” where the first line represents the type of sorting algorithm(1,2,3,4 as above), the second line represents the number of workers, and the following lines correspond to the speeds of the workers.

Sample “hw2.inp” :

```
3
4
10
1
5
2
```

OUTPUT Specifications

Your code should generate a file named "hw2.out", where first line is the total minimum time that is required for all workers to get out of the mine. In the subsequent lines, you should give the strategy to achieve this specified minimum time, where these lines can contain either one or two integers, indicating which worker(s) form the next group to pass (or turn back). Use space as a delimiter in each line. As the final line you should output the execution times of the specified sorting algorithm. The execution times must be printed in milliseconds.

- Sample "output.txt" :

$t_{\text{quicksort}}$	← Time required for quicksort, since 3 is given in input
17	← Total minimum time
1 2	← Workers with passing time 1 and 2 get out of the mine
1	← Worker with passing time 1 takes the torch back
5 10	← Workers with passing times 5 and 10 get out of the mine
2	← Worker with passing time 2 takes the torch back
1 2	← Workers with passing times get out of the mine

Written part

You are expected to bring in a short report. Run each sorting algorithm for the arrays having {50, 100, 500, 1000, 5000, 10000} elements(N). Then, plot a chart with axes **running time of sorting algorithms** vs **N** (When plotting run the algorithms on the same input) for each sorting method you use. Plot those (4) charts on one single chart (using Excel or OpenOffice Calc) and add a paragraph explaining the details (*both explanation and pseudocode*) of your **sorting algorithm** in your report.

Sample code for time measurement

Please use this code for calculating the running times.

```
#include <sys/time.h>
...
struct timeval tv_begin;
struct timeval tv_end;
int passed_milliseconds;

gettimeofday(&tv_begin, NULL);

sort();

gettimeofday(&tv_end, NULL);

passed_milliseconds = (tv_end.tv_sec - tv_begin.tv_sec)*1000 + (tv_end.tv_usec - tv_begin.tv_usec)/1000;
```

Submission

- Upload hw2.c or hw2.cpp files to COW until 11th of November, 23:59. Bring the written report to B-204 until 17:00 the next day (12th of Nov. , Monday).
- All work should be done individually. In case you cheat, you will get 0 from your homework and disciplinary action will be taken.