

1.204 Computer Algorithms in Systems Engineering

Spring 2010

Problem Set 5: Project management

Due: 12 noon, Wednesday, April 7, 2010

1. Problem statement

The most widely used scheduling technique is the critical path method (CPM) for scheduling, often referred to as *critical path scheduling*. This method calculates the minimum completion time for a project along with the possible start and finish times for the project activities. The *critical path* represents the set or sequence of predecessor/successor activities which will take the longest time to complete. The duration of the critical path is the sum of the activities' durations along the path. Thus, the critical path can be defined as the longest possible path through the "network" of project activities. The duration of the critical path represents the minimum time required to complete a project. Any delays along the critical path imply that additional time would be required to complete the project.

Table 1 gives an example project:

TABLE 1 Precedence Relations and Durations for a Nine Activity Project Example

Activity	Description	Predecessors	Duration
A	Site clearing	---	4
B	Removal of trees	---	3
C	General excavation	A	8
D	Grading general area	A	7
E	Excavation for trenches	B, C	9
F	Placing formwork and reinforcement for concrete	B, C	12
G	Installing sewer lines	D, E	2
H	Installing other utilities	D, E	5
I	Pouring concrete	F, G	6

The critical path schedule solution for this project is given below:

Activity	Duration D(i)	Earliest start time E(i)	Latest finish time LF(i)	Latest start time LS(i)
A	4	0*	4*	0
B	3	0	12	9
C	8	4*	12*	4
D	7	4	22	15
E	9	12	22	13
F	12	12*	24*	12
G	2	21	24	22
H	5	21	30	25

I	6	24	30*	24
---	---	----	-----	----

*Activity on a critical path since $E(i) + D(i) = LF(i)$.

Each task has an earliest start time $E(i)$. Activities with no predecessor have an earliest start time = 0. Other activities have an earliest start time equal to the latest completion time of any predecessors. Each task also has a latest start time $LS(i)$, which is the time such that $LS(i) + D(i) = LS(j)$ where j is a successor to i . This is equivalent to latest finish time $LF(i) = LS(j)$ where j is a successor to i . See http://pmbok.ce.cmu.edu/10_Fundamental_Scheduling_Procedures.html for background.

Note: The critical path method assumes that all resources are available without constraint. If activities require resources and there is a fixed limit, this makes this into an integer programming problem. For example, if tasks A through I each require a number of workers, but there is a fixed number of workers, the critical path schedule may exceed the number of available workers at times. Commercial project management software often uses heuristics to solve resource-constrained schedules. A few systems use integer programming, typically branch and bound or dynamic programming.

2. Assignment

Modify one or more of the algorithms and data structures we have covered in class to solve the critical path scheduling problem. At a minimum, output the critical path tasks: their sequence, durations and the length of the critical (minimum) path. Optionally output the earliest start time and latest finish time for each task. Use any convenient method for input and output.

Turn In

1. Place a comment with your full name, Stellar username, and assignment number at the beginning of all `.java` files in your solution.
2. Place all of the files in your solution in a single zip file.
 - a. Do not turn in electronic or printed copies of compiled byte code (`.class` files) or backup source code (`.java~` files)
 - b. Do not turn in printed copies of your solution.
3. Submit this single zip file on the 1.204 Web site under the appropriate problem set number. For directions see **How To: Submit Homework** on the 1.204 Web site.
4. Your solution is due at noon. Your uploaded files should have a timestamp of no later than noon on the due date.
5. After you submit your solution, please recheck that you submitted your `.java` file. If you submitted your `.class` file, you will receive **zero credit**.

Penalties

- 30 points off if you turn in your problem set after Wednesday noon but before noon on the following Monday. You have one no-penalty late submission per term for a turn-in after Wednesday noon and before Monday noon.
- No credit if you turn in your problem set after noon on the following Monday.

MIT OpenCourseWare
<http://ocw.mit.edu>

1.204 Computer Algorithms in Systems Engineering
Spring 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.