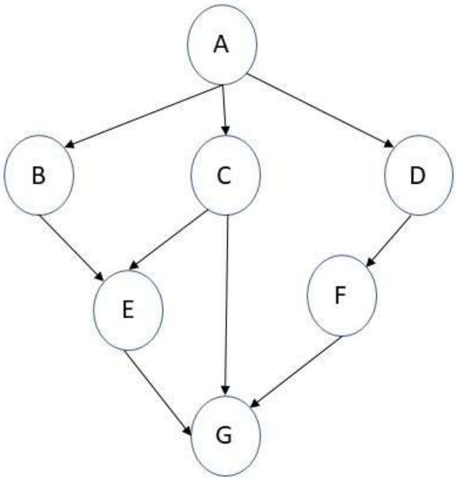
Multicore Processors: Architecture & Programming

Homework Assignment 1

[Total 40 points]

1. Assume we have the following task graph. An arrow from a task to another means that the first task generates data needed by the second one. Do not make any assumptions regarding cache misses, coherence, etc. The table indicates the time taken by each task. A task cannot be decomposed to smaller tasks.



Task#	Duration
A	10
B	50
C	100
D	100
E	50
F	50
G	10

a. [2 points] What is the smallest number of cores needed to get the best performance? Show the steps that you did to reach your conclusion.

b. [1 points] Based on the number of cores you picked in part (a) above, what is the total run time?

c. [2 points] What is the span for the above graph? what is the work?

d. [2 points] What is the parallelism of that graph?

e. [1 points] What does the number you calculated in (d) mean?

2. [4] From Flynn classification, it is obvious that MIMD can execute programs that fit in the other three types. However, companies are designing chips based on SIMD (e.g. GPUs). Why is that? State two reasons.

3. [3] If you are given a sequential program that you are required to parallelize, first you need to find the parts that are *parallelizable.* However, in some cases, it is not worth it to parallelize those parts, why?

4. [6] Suppose you have two parallel programs that solve the same problem and both programs are correct. State three factors that you consider when comparing the two programs and see which one is better. Do not consider price of the software for this question. State at least three other factors,

5. [5] Suppose, for a specific problem, we know the best sequential algorithm for it for a single core. Does this mean that this algorithm is also the best for multicore? Justify with at least two reasons.

6. Suppose we have the algorithm (assume N is a large even number):

```
for(i = 0; i < N/2; i++)
        a[i] += a[i+ N/2 ];
```

a. [4] Can we parallelize the above algorithm? If no, why not? If yes, explain.
b. [4] What is the maximum number of cores after which no performance enhancement. can be seen? Justify [Note: that number can be one if you answered "No" in part a.]

7. A program with 20% part that must be executed sequentially, is required to be accelerated three-fold.
How many cores are required for this task [2]?
How about five-fold speedup [2]?
Show all steps in the above two questions.

8. [2] Suppose we have a system with two levels of caches: L1 and L2. Level 1 is close to the processor, level 2 is the last level before accessing the main memory. We know that two main characteristics of a cache performance are: cache access latency (How long does the cache take before responding with hit or miss?) and cache hit rate (how many of the cache accesses are hits?). As we go from L1 to L2, which of the two characteristics become more important? and why?