

## Homework 2

---

Problem 1 (#10 in text on p. 111):

The Red/Blue computation simulates two interactive flows. An  $n \times n$  board is initialized so cells have one of three colors: red, white, and blue, where white is empty, red moves right, and blue moves down. Colors wrap around on the opposite side when reaching the edge.

In the first half step of an iteration, any red color can move right one cell if the cell to the right is unoccupied (white). On the second half step, any blue color can move down one cell if the cell below it is unoccupied. The case where red vacates a cell (first half) and blue moves into it (second half) is okay.

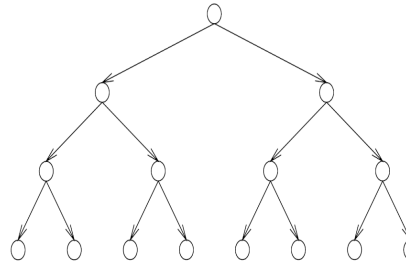
Viewing the board as overlaid with  $t \times t$  tiles (where  $t$  divides  $n$  evenly), the computation terminates if any tile's colored squares are more than  $c\%$  one color. Use Peril-L to write a solution to the Red/Blue computation.

## Homework 2, cont.

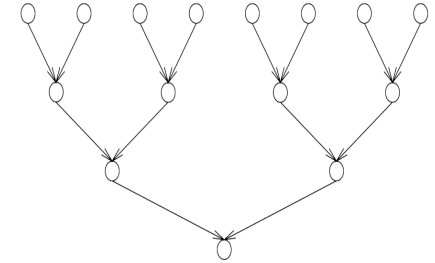
Problem 2:

For the following task graphs, determine the following:

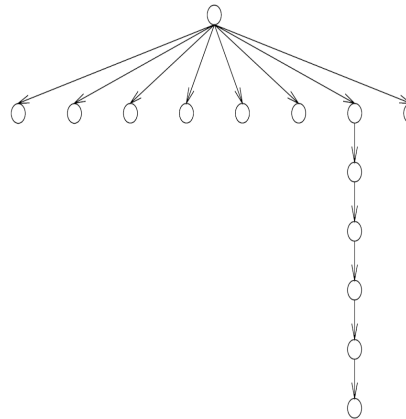
- (1) Maximum degree of concurrency.
- (2) Critical path length.
- (3) Maximum achievable speedup over one process assuming that an arbitrarily large number of processes is available.
- (4) The minimum number of processes needed to obtain the maximum possible speedup.
- (5) The maximum achievable speedup if the number of processes is limited to (a) 2 and (b) 8.



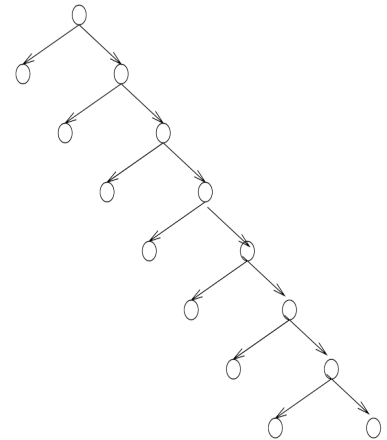
(a)



(b)



(c)



(d)

Figure 3.42 Task-dependency graphs for Problem 3.2.