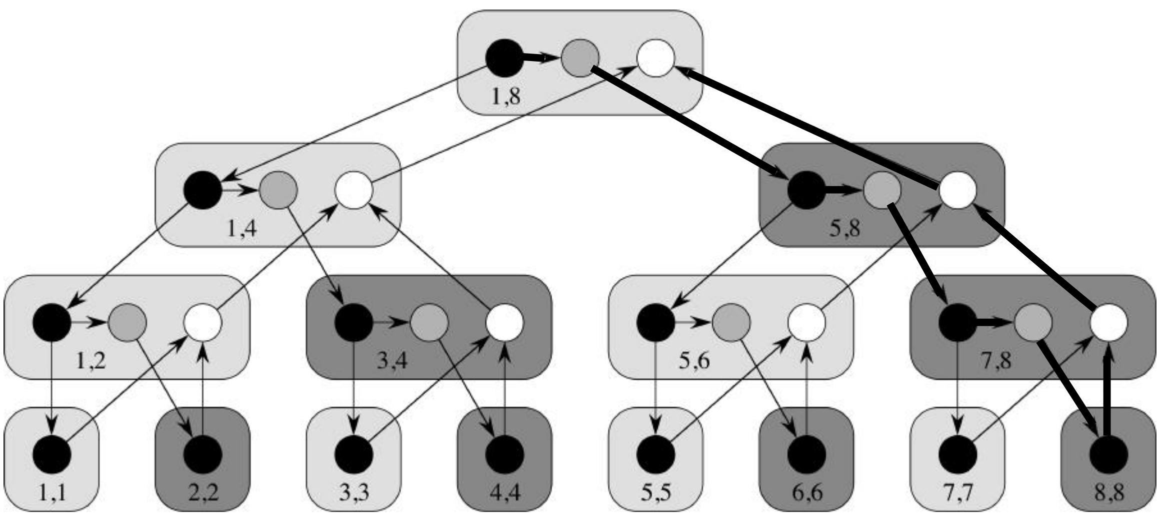
Name: ____

1. (4 points) The algorithm P-SUM(L, p, r) computes the sum of the elements $L[p \mathrel{.\,.} r]$ of an array L of length n. Draw the DAG for P-SUM$(L, 1, 8)$ where L is an array of length 8. Determine the work, span, and parallelism from the DAG. Show all work.

P-SUM(L, p, r)

if $p < r$
    $q = \lfloor (p + r)/2 \rfloor$
    $x =$ **spawn** P-SUM(L, p, q)
    $y =$ P-SUM$(L, q + 1, r)$
    sync
    return $x + y$
else
    return $L[p]$

Solution



The critical path is bolded. The span is the length of the critical path (in strands, not edges), so $T_\infty = 10$. The work is the number of strands, so $T_1 = 29$.

The parallelism is $T_1/T_\infty = 2.9$.

(continued on other side)

2. (4 points) The algorithm P-TRANSPOSE computes the transpose of an n-by-n matrix A in place.

```
P-TRANSPOSE(A)
1  n = A.rows
2  parallel for j = 2 to n
3      for new i = 1 to j − 1
4          exchange a_ij with a_ji
```

Determine the work, span, and parallelism of P-TRANSPOSE. What is the *parallel slackness* when $n = 256$ and $P = 16$? Justify your answers.

Solution

The work is just the serial running time of the algorithm. The loop on lines 3 – 4 is $\Theta(j)$ since exchanging two elements is a $\Theta(1)$ operation. Then the total running time of the nested loops is

$$\sum_{j=2}^{n} \Theta(j) = \Theta\left(\sum_{j=2}^{n} j\right) = \Theta\left(\frac{n(n+1)}{2} - 1\right) = \Theta(n^2).$$

The other work (line 1) is constant time, so the work is $T_1(n) = \Theta(n^2)$.

As noted above, the running time for the inner loop (which is not parallelized) is $\Theta(j)$. Using the formula for the span of a parallel loop, we have

$$T_\infty(n) = \Theta(\lg n) + \max_{2 \le j \le n} \text{iter}_\infty(j).$$

But $\text{iter}_\infty(j) = \Theta(j)$ and $\max_{2 \le j \le n} \Theta(j) = \Theta(n)$, so we conclude that $T_\infty(n) = \Theta(\lg n) + \Theta(n) = \Theta(n)$. It follows that the parallelism is

$$T_1(n)/T_\infty(n) = \Theta(n^2)/\Theta(n) = \Theta(n).$$

The parallel slackness when $n = 256$ and $P = 16$ is

$$\frac{T_1(n)}{P \cdot T_\infty(n)} = \frac{256^2}{16 \cdot 256} = 16.$$

3. (2 points) The function MAT-VEC(A, x) computes the matrix-vector product $y = A \cdot x$ where the components y_i of the vector y are given by

$$y_i = \sum_{j=1}^{n} a_{ij} \cdot x_j.$$

The implementation includes a *race condition*. Locate and describe the race condition and indicate how the code should be modified to fix the problem.

```
MAT-VEC(A, x)
1  n = A.rows
2  let y be a new vector of length n
3  parallel for i = 1 to n
4      y_i = 0
5  parallel for i = 1 to n
6      for j = 1 to n
7          y_i = y_i + a_ij · x_j
8  return y
```

Solution

The problem is the lack of "**new** j" in the loop on line 6. The parallel for loop on line 5 may create a separate thread for each value of i, with each thread executing lines 6 – 7 *with a common j variable*. Adding the **new** keyword will fix the problem.