Name: ____

The exam consists of seven problems; you only need to solve five. You *must* do problems 1, 2, and 3; the remaining two may be chosen from problems 4 – 7. Please indicate the problems you want to have graded by circling the problem numbers — otherwise, I will just grade the first four problems you worked on.

The following reference materials are provided on the last page of the exam: the statement of the Master Theorem and some summation formulas.

- You **have** 120 minutes.
- You **may use only** a calculator, pencil, or pen; you may not use a phone as a calculator.
- You **must** show all calculations. If you fail to show your work, you will receive no credit.
- You **must** put away all notes and papers and close all bags.

1. **(REQUIRED)** Consider the following algorithm to multiply two n-bit numbers x and y:

Recursive-Multiply(x, y)

```
n = x.size
if n == 1
    return xy
else
    Write x = x_1 · 2^{n/2} + x_0 and y = y_1 · 2^{n/2} + y_0
    p = Recursive-Multiply(x_1 + x_0, y_1 + y_0)
    q = Recursive-Multiply(x_1, y_1)
    r = Recursive-Multiply(x_0, y_0)
    return q · 2^n + (p − q − r) · 2^{n/2} + r
```

(a) Derive a recursion for the running-time of Recursive-Multiply.

(b) Solve the recursion to find an asymptotic bound for the running-time.

(c) Show that Recursive-Multiply is equivalent to the 'schoolbook' method of multiplication.

2. **(REQUIRED)** Consider the following recursive function which computes the minimum value in an array x of length $x.\mathit{length}$:

```
RECURSIVE-MIN(x)
n = x.length
if n == 1
      return x[1]
else
      p = RECURSIVE-MIN(x[2 .. n])
      return MIN(x[1], p)          // two-argument MIN() function
```

(*a*) Derive a recursion for the running-time of RECURSIVE-MIN.

(*b*) Use a recursion tree to 'guess' an asymptotic bound for the recursion.

(*c*) Use the substitution method to prove your guess is correct.

3. **(REQUIRED)** Consider the following variant of the subset-sum problem. There are n items, each having non-negative, integer weight $w_i, i = 1, 2, \ldots, n$. We wish to select a set S of items such that

$$\sum_{i \in S} w_i \leq W$$

where W is a non-negative, integer bound. Moreover, we want S to maximize the sum, subject to the constraint.

(a) Show that this problem has optimal substructure.

(b) Consider the following greedy approach: at each step, add to S the item of largest weight that maintains the bound ($\sum_{i \in S} w_i \leq W$). Construct an example that shows that this greedy approach does not necessarily produce an optimal solution.

4. The algorithm P-Sum computes the sum of the elements of an array L of length n:

```
P-Sum(L)
n = L.length
if n == 1
    return L[1]
c = ⌊n/2⌋
x = spawn P-Sum(L[1 .. c])
y = P-Sum(L[c + 1 .. n])
sync
return x + y
```

(*a*) Draw the computation DAG of P-Sum(L) for L of length 4.

(*b*) Identify the critical path on the DAG; determine the span.

(*c*) Determine the work and parallelism of P-Sum(L) with L of length 4.

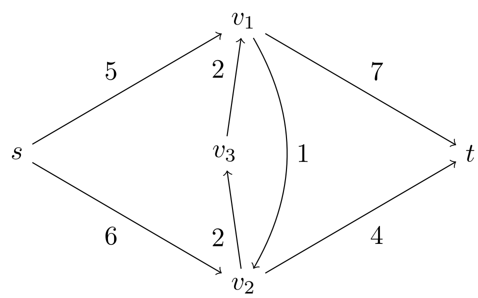
5. Consider an RSA key set with $p = 17$, $q = 19$, $n = 323$, and $e = 5$.

   (a) Use the Extended Euclidean Algorithm to find the decryption exponent d.

   (b) Show that $a = 2$ is *not* a witness to the compositeness of q.

   (c) Verify that $S = 4$ is a valid signature for the message $M = 55$.

6. Consider the following flow graph G:



(a) Show the execution of the Edmonds-Karp algorithm on G. For each iteration, show the flow f on the graph G, the residual graph G_f, and the augmenting path. Include all backwards flows in the residual graph.

(b) What is the minimum cut corresponding to the maximum flow on this network?

7. Let G be an undirected graph, u and v vertices of G, and k a non-negative integer.
Define Longest-Path-Length(G, u, v) to be the optimization problem of determining the length of the longest simple path from u to v in G, and Longest-Path(G, u, v, k) the decision problem "is there a simple path from u to v in G of length at least k?" Show that Longest-Path $\in P$ if and only if Longest-Path-Length is solvable in polynomial time.

Theorem (Summation Formulas)**.**

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{i=0}^{n} r^i = \frac{1-r^{n+1}}{1-r}$$

Theorem (Master Theorem)**.** *Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence*

$$T(n) = aT(n/b) + f(n)$$

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

(a) *If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.*

(b) *If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$.*

(c) *If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$.*