Name: ____________________________________

| | |
|---|---|
| 1 | /20 |
| 2 | / 8 |
| 3 | /17 |
| 4 | /15 |
| 5 | / 8 |
| total | /68 |

# CSE 421
# Final Exam
March 18 , 2002

**Instructions:**

- You have 1 hour and 50 minutes to complete the exam.
- Please do not turn the page until you are instructed to do so.
- Good luck!

1. (20 points, 2 each) Indicate for each of the following if it is **true or false** or in some cases **not-known** by circling the appropriate answer.

   - *True or False:* Suppose that we have the recurrence $f(i) = max(i + f(i-4), 2i + f(i-2))$, and we know that $f(i) = 0$ for $i = 1, 2, 3, 4$. Then dynamic programming allows us to calculate $f(n)$ in time $O(n)$.
   - *True or False:* Let $S$ be a set of elements. Suppose that we have the recurrence $OPT(S) = min_{s \in S}(OPT(S - \{s\}) + w(s))$, where $w(s)$ is the weight of $s$, and $OPT(\emptyset) = 0$. Then dynamic programming allows us to calculate $OPT(\{1, 2, \ldots, n\})$ in time $O(n^2)$.
   - *True or False:* Consider the problem of shortest paths in a graph $G$ where edges can have negative weights. Recall that we defined $Opt(i, v)$ to be the length of the shortest path from $s$ to $v$ that uses at most $i$ edges. Suppose that there is a some vertex $w$ such that $Opt(n, w) \neq Opt(n-1, w)$ (where $n$ is the number of nodes in the graph). Then $G$ has a negative cycle.
   - *True or False:* Same setup as previous question. Suppose that $Opt(1, v) = Opt(2, v)$ for all $v$ (in a graph where there are $n > 2$ vertices). Then $G$ has no negative cycle reachable from $s$.
   - *True or False:* There is no valid circulation in a graph that has a demand of 1 at every node.
   - *True or False or Unknown:* We know of a problem in $\mathcal{NP}$ that is also in $\mathcal{P}$.
   - *True or False or Unknown:* Suppose that $X$ and $Y$ are both in $\mathcal{P}$. Then there is a polytime reduction from $X$ to $Y$.
   - *True or False or Unknown:* Suppose that $X \leq_{\mathcal{P}} Y$. Then $Y \leq_{\mathcal{P}} X$.
   - *True or False:* Suppose that $X \leq_{\mathcal{P}} Y$, $X$ is NP-complete and $Y \in \mathcal{NP}$. Then $Y \leq_{\mathcal{P}} X$.
   - *True or False or Unknown:* Suppose that $SAT \leq_{\mathcal{P}} X$ and $SAT \leq_{\mathcal{P}} Y$. Then $X \leq_{\mathcal{P}} Y$.

2. (8 points, 2 each) Indicate for each of the following if it is **true or false** by circling the appropriate answer. In all of the following, you are given an s-t flow network $G$, where $c(u, v)$ is the capacity of edge $(u, v)$.

   - *True or False:* Let $f$ be a maximum flow in $G$, where $f(u, v)$ is the flow on edge $(u, v)$. Let $(A_1, B_1)$ and $(A_2, B_2)$ be two minimum s-t cuts. Then $\sum_{u \in A_1, v \in B_1} f(u, v) = \sum_{u \in A_2, v \in B_2} c(u, v)$.
   - *True or False:* Let $f$ be a maximum flow in $G$. Let $(A_1, B_1)$ and $(A_2, B_2)$ be two minimum s-t cuts. Then $\sum_{u \in B_1, v \in A_1,} c(u, v) = \sum_{u \in B_2, v \in A_2} c(u, v)$. (Pay close attention to the subscripts.)
   - *True or False:* Let $f$ be a maximum flow in $G$. Let $(A_1, B_1)$ be a minimum s-t cut. Then $\sum_{u \in B_1, v \in A_1} f(u, v) = 0$.
   - *True or False:* Suppose there is an s-t cut in $G$ with total capacity $C$. Then there is a flow of value $C$ in $G$.

3. (17 points) Consider the following flow network, with a flow $f$ shown. An edge labelled with "$a/b$" means that the flow on that edge is $a$ and the capacity of the edge is $b$.

(a) (4 points) Draw the residual graph $G_f$. Indicate each edge (with its directionality) in the residual graph and, next to it, its residual capacity.

(b) (4 points) What augmenting path has the maximum bottleneck capacity? (Specify the names of the vertices on the path in order.) What is the bottleneck capacity of this path?

(c) (4 points) Indicate the new flow on each edge on the following diagram *after* augmenting along the path specified in step 2.

(d) (5 points) Give an example of a circulation with demands problem with $\sum_v d_v = 0$ where there is no feasible solution. Your circulation problem should fit in the space given. Give your example by drawing the flow network, indicating the capacities on each edge and the demands of each node. Please try to make your example as small as possible.

4. (15 points) In a word processor, the goal of "pretty-printing" is to take text with a ragged right margin — like this:

```
Call me Ishmael.
Some years ago,
never mind how long precisely,
having little or no money in my purse,
and nothing particular to interest me on shore,
```

— and turn it into text whose right margin is as "even" as possible — like this:

```
Call me Ishmael.  Some years ago, never
mind how long precisely, having little
or no money in my purse, and nothing
particular to interest me on shore,
```

We formulate the problem of pretty-printing as follows. Suppose our text consists of a sequence of *words* $W = \{w_1, w_2, \ldots, w_n\}$.

A *formatting* of $W$ consists of a partition of the words in $W$ into *lines*. Let $B_{i,j}$ be the "badness" of putting the sequence of words $w_i, w_{i+1}, \ldots, w_j$ on a single line. (For example, $B_{i,j} = \infty$ if these words don't fit on a line. Otherwise, it is a measure of how much space is left over at the right margin. Thus, we want to choose lines that have small badness values.)

Specifically, our goal is to construct a partition of a set of words $W$ into lines so as to minimize the sum of the *squares* of the badness of all lines. For example, if $W = w_1, \ldots, w_n$ are partitioned into two lines, say $w_1, \ldots w_i$ and $w_{i+1}, \ldots w_n$, then the value of the solution is $B_{1,i}^2 + B_{i+1,n}^2$. But there may be a different partition (possibly into a different number of lines) that has smaller value.

- (8 points) Give a recurrence for $OPT[i]$, the value of the optimal solution on the set of words $W_i = \{w_1, \ldots, w_i\}$. You may assume that $B_{i,j}$ for all $i \leq j$ is part of the input. You do not need to explain your recurrence. Be sure to specify a base case.

- (4 points) What is the running time (in big Oh notation) of the dynamic programming algorithm that determines the value of the optimal solution for $W_n$? You do not need to give the algorithm.

- (3 points) What is the running time (in big Oh notation) of the dynamic programming algorithm that determines the actual partition in the optimal solution for $W_n$? You do not need to give the algorithm.

5. (8 points)

You are given an array $A[1..n]$ of values from some universe. Each pair of array values are either the same or not. The only feasible operation on these values that you can do is to pick two array entries and perform a constant-time test of the form "$A[i] = A[j]$?". Your goal is to design an algorithm that determines if there is any value that occurs strictly more than $n/2$ times in the array.

Your basic recursive procedure

`Solve(`$A[i, .., j]$`)`

returns a pair $(b, v)$ such that $b$ is "yes" if there is a value that occurs strictly more than $(j - i + 1)/2$ times in the input array and "no" otherwise, and $v$ is the value that occurs in a majority of array entries if there is one.

Fill in the missing portion of the following high-level psuedocode for a divide and conquer solution to this problem. Your solution should be such that the resulting algorithm runs in $O(n \log n)$ time. Also your solution should fit on this page. No explanations are needed.

```
(b1, v1) :=Solve(A[1..n]);
    begin
    If n = 1 return (yes, A[1]);
    If n = 2
        test if the two values are equal ;
        return (yes, A[1]) if they are equal,
                or (no, X) otherwise.
    (b1, v1) :=Solve(A[1..⌊n/2⌋]);
    (b2, v2) :=Solve(A[⌊n/2⌋ + 1..n]);




    end
```