Review

Final exam:
- Thursday, Feb 25, 2:45-4:45pm, room 70-2400
- one letter-size double-sided cheat sheet: handwritten

Format of the exam:
- Short answers: multiple choice, trace a Python/pseudo code, apply algorithm from class, estimate time complexity
- Pseudo code: fix, trace (including stack-frame diagram)
- Lab like problems: design your own pseudo code, estimate time complexity, provided test cases

Announcements: lab 9 (project reflection) – due tonight
final project – due tomorrow 2pm
Q1:
Implementing queue, stack, or neither via linked-lists:

a. Insert at front, remove from front
b. Insert at rear, remove from rear
c. Insert at rear, remove from front
d. Insert at front, remove from rear
Q2:

Another operation(s) for stack/queue:

a. Get size
b. Test if empty
c. Sort
d. Extract value from the middle of the list
e. Insert value to the middle of the list
Q3:

Running times of sorting algorithms, choose from $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(2^n)$

MergeSort

QuickSort

HeapSort

A faster algorithm for data that is already sorted?

$O(n)$ bubble sort for

in the worst-case

(and most of the time): $O(n^2)$
Q4:
Show the order in which vertices are visited when searching for G from A. Assume neighbors are processed alphabetically.
Q5:
Subset Sum problem: for a capacity $C$ and a list of items, each item has a specified weight, find if there are items with total weight = $C$.

Solving via backtracking:
- What is an interim configuration? 
- How to get from one config to the next?
- What is a goal config? 
- What is a failure config?
Q6:
Sudoku board with 50 already-assigned cells. How many final boards are there, valid or not?
Q7:
Heapify the following binary tree:

Then, use the heap to HeapSort the data.
Q8:

In a complete binary tree stores in an array, what are the formulae for:

parent(i):

left-child(i):

right-child(i):
Q9:
Representing the board in the Labyrinth game in a hash table that maps keys representing pairs of coordinates to the necessary info stored at that location on the board:

- What is a key in this example?

- What is a value?

- Suggest a hash fnc minimizing collisions.
Q10:
Sort the following data using MergeSort, QuickSort (always choose the first value as the pivot), and HeapSort. Show each step for lists > 3 elements.

6, 7, 8, 9, 5, 4, 3, 2, 1
Q11:
Trace the jobSchedule(joblist) and DPjobSchedule(joblist) pseudo codes from class (pseudo code is provided) on this sample data:
Format: (job, start, end, pay)

(A, 1, 2, 10), (B, 2, 4, 25), (C, 2, 3, 10), (D, 3, 5, 10)
Q12:
Trace Dijkstra from A to all other vertices. Indicate:
- final distance and predecessor for every vertex
- the order in which vertices are finalized
- the shortest path from A to C
Review

Q13:
Consider circularly linked list.
- pseudo code: append( headNode, value )

- time complexity for a list of length n:
- testing:
Q13 cont.:
Consider circularly linked list.
- pseudo code: maxNode( headNode )

- time complexity for a list of length n:
- testing:
Q14:

Grocery store with aisles 0..n-1, every Item object has a name and an aisle, e.g. Item('milk',4). We have a list of items in the store (inventory) and a list of things to buy (groceryList).

- pseudo code: organize(inventory, groceryList) returns a list of things to buy for every aisle. Describe data structures.

- time complexity:

- testing:
Q15:

Backtracking skeleton:

```python
def findSol(c):
    ...
    result = None
    for each new config d one step away from c:
        solution = findSol(d)
        ??? (choose from the provided choices)
    ...
    return result
```
Review

Note:

Project:
- Sectional tournament: Thursday 2-4pm
- Battle Royal: Saturday 1pm in 70-1400 big auditorium

Final exam reviews:
- David’s SI session this week: Friday 2-4pm
- Gabbie and David will hold a review session on Wednesday, Feb 24, 2010, 6pm in room 70-3660. There will be SSE review session, date/time/room to be announced

GOOD LUCK on all your finals!