CS-510 Algorithms **Problem Set - Randomized Algorithms**

Problem 1. Design a randomized algorithm for the MAT SAT problem, i.e. given a Boolean formula on n variables and with m clauses. There is no restriction on the number of literals in each clause.

Problem 2. Design a Las Vegas algorithm for the MAX 3-SAT problem, that satisfies at least $7m/8$ clauses. What is the expected runtime of your algorithm.

Problem 3. Let A be an array of n distinct numbers. Suppose A contains the numbers from 1 to n and is filled by taking a uniform random permutation of $[1...n]$. Use indicator random variables and linearity of expectation to compute the expected number of inversions in A.

Problem 4. Suppose we run the randomized minimum cut algorithm that we studied in class on a graph G which has no cycles. If G is disconnected, then the minimum cut is of size 0 (we can determine it by running a BFS or DFS). Suppose G is connected, that is G is a tree. Show that the randomized algorithm will output a minimum cut with probability 1. Note that in a tree it is very easy to find a min cut (just get a leaf on one side of the cut and the rest of the tree on the other).

Problem 5. In the class we discussed a randomized algorithm for the Max 3-SAT problem. We got that setting each of the n variables independently equal to 1 or 0 with probability 1/2, gives us that the expected number of satisfied clauses is at least $7m/8$, where m is the number of clauses. This however had an underlying assumption that no variable and its negation appears in the same clause.

Show that even if we allow a variable and its negation appear in the same clause, the above algorithm will still satisfy at least $7m/8$ clauses in expectation.

Problem 6. Design a randomized algorithm for the MAT SAT problem, i.e. given a Boolean formula on n variables and with m clauses. There is no restriction on the number of literals in each clause.

Problem 7. In class we discussed a greedy 2-approximate algorithm for the Max cut problem. Design a randomized algorithm for the problem, such that the expected value of the cut produced by your algorithm is at least $OPT/2$, where OPT is the value of the maximum cut.

Problem 8. Consider the following algorithm for searching a given number x in an unsorted array $A[1,\ldots,n]$ having n distinct values.

Algorithm 1 Randomized Search

```
Choose any i uniformly at random from 1,...,n
if A[i] = x then
    return i
else
    repeat
```

1. Assume that x is present in A, what is the expected number of comparisons made by the algorithm before it terminates.
2. Suppose A does not have distinct values and there are exactly 2 indices i such that $A[i] = x$. What is the expected number of comparisons made by the algorithm before it terminates.

Problem 9. Closest Pair Problem

Given a set of points $S = p_1, p_2, ..., p_n$ in the plane. Give a randomized algorithm to find the pair of points p_i, p_j that are closest together. Analyze your algorithm and prove that it runs in $O(n)$ time.