

2002 Fall MSCS Exam – Algorithms
Answer one question from each of the two groups.

Group 1 – Answer one of these two

1. Suppose we have to do some string matching; we have a string of letters from input, and a pattern that we wish to match against. In our matching pattern, the “*” matches zero or more arbitrary characters, and the “.” matches only a single arbitrary character. We could match “XYABCZQABMMC” with “XY*Z.AB*C,” where the first * would match “ABC”, the period matches “Q”, and the second * matches “MM”.

Sketch pseudocode for an efficient algorithm that determines if a string matches a pattern.

2. Consider a recursive binary search algorithm for finding a number in a sorted array. The code might look something like the following.

```
int binsearch(int A[], int p, int q, int query)
{
    split = (p + q)/2
    if (A[split] == query) return true;
    if (p == q) return false;
    if (A[split] < query) return binsearch(A, p, split, query);
    else return binsearch(A, split + 1, q, query);
}
```

Normally, we pass a pointer to the array, making the time required to pass the arguments $O(1)$. What is the complexity of the algorithm above? Suppose instead that we push a copy of the entire array for each call; what is the complexity then? And if we push only the portion of the array that we need– what is the complexity for this variation?

Group 2 – Answer one of these two

1. Describe the differences between the Ford-Fulkerson method, and the Edmonds-Karp algorithm. Why might you prefer one to the other? In what circumstances would one of these perform poorly?
2. Given a graph, describe an algorithm to find the maximum number of edge-disjoint paths between a pair of vertices S and T .