Romerico David, Haley Elliott, Julian Halsey, Nkechiyem Molokwu
COSC 336
04/09/2024

Assignment 5

Programming Task:

Description of Algorithm:

Our algorithm determines the maximum sum of increasing subsequences for a given integer array through dynamic programming. We initialize two arrays: s[0...n] to store the maximum sum of an increasing subsequence ending at each index i, and p[0...n] to store the index of the preceding element in that subsequence. Since the only increasing subsequence that can be formed at the first element is just itself with no predecessor, s[0] = a[0] and p[0] = -1. Then, we iterate through the entire array, updating s[i] and p[i] as we determine the maximum sum at each index. After we find all values of s[0...n], we find the index of the maximum sum in s[0...n], which serves as a reference point to reconstruct the increasing subsequence by iterating backwards through p[0...n].

Code:

```
public static int maxSumLIS(int[] a) {
    if (a.length == 0)
        return 0;

    int n = a.length;
    int[] s = new int[n];
    int[] p = new int[n];

    s[0] = a[0];
    p[0] = -1;

    for (int i = 1; i < n; i++) {
        s[i] = a[i];
        p[i] = -1;
        for (int j = 0; j < i; j++) {
```

```
                if (a[i] >= a[j] && s[i] <= s[j] + a[i]) {
                    s[i] = s[j] + a[i];
                    p[i] = j;
                }
            }
        }

        int maxIndex = 0;
        for (int i = 1; i < n; i++) {
            if (s[i] > s[maxIndex])
                maxIndex = i;
        }

        int[] subsequence = new int[n];
        int subIndex = 0;
        for (int i = maxIndex; i != -1; i = p[i])
            subsequence[subIndex++] = a[i];

        System.out.print("Increasing subsequence with the max sum: ");
        for (int i = subIndex - 1; i >= 0; i--)
            System.out.print(subsequence[i] + " ");
        System.out.println();

        return s[maxIndex];
    }
```

Results Table:

Data Set #	Max Sum	Subsequence
5.3	130,021	4601, 20255, 23073, 32092, 50000
5.4	143,418	25197, 26355, 29960, 30953, 30953

Table 1: Programming Task 1