Review sheet for Final Exam (Dec. 15 Thursday, 8:30-10:00a)

**Basic skills:**

run time analysis- Analysis of the time and space complexity of an algorithm. This course only focuses on time complexity which relates the input size of a function to a number of computations.

big-O notation- Asymptotic upper bound of a function described in terms of the input size n. In CS terms this is the worst case number of computations that has to be performed for a given input size. As the input size n approaches infinity, the highest order term dominates the rest so the big-O notation can be reduced to just that term, e.g. 6n^2+4n+100 = O(n^2)

**Chapter 5:**

Manhattan tourist problem-

In: A weighted n x m graph

Out: Path with maximum-weight from source (0,0) to sink (n,m). Can only move down and to the right.

The longest path in a directed graph problem-

In: An edge-weighted directed graph with source and sink nodes

Out: A longest path from source to sink in the graph

The Dynamic programming algorithm-

The change problem and DP algorithm-

DP for sequence alignment-

Exercising DP in the Alignment graph-

Scoring matrix for sequence alignment-

Global alignment and local alignment-

DP algorithm for sequencing alignment using affine gap penalty-

Divide-and-Conquer algorithm for sequence alignment using linear space-

Link to completed Dynamic Programming solution for sequence alignment:

<https://codeshare.io/298MBG>

**Chapter 6:**

Random breakage model:

There are fragile regions in genomes where genome rearrangements occur most frequently.

Genome rearrangements: where DNA fragments are reversed and otherwise rearranged

Identity Permutation: this is just the natural order of the numbers, (e.g 1 2 3 4 5 6 … n)

Sorting by reversals: Find the distance between any given permutation (rearranged genome) and the

Identity permutation

Reversal distances-

Greedy algorithm for sorting reversals-

Breakpoint theorem-

Translocations-

fusion and fissions-

2-break distance problem-

Breakpoint graph-

2-break distance theorem-

Fragile breakage model-

Synteny block construction-

**Chapter 7:**

Evolutionary tree-

Distance-based phylogeny construction-

additive matrix-

four-point condition-

Distance between leaves problem-

Limb-length theorem-

LimbLength(i) is equal to the minimum value of (Di,k + Di,j – Dj,k)/2 over all leaves j and k.

Additive phylogeny-

Least-square phylogeny-

Ultrametric trees-

UPGMA-

Neighbor-joining algorithm-

Character-based phylogeny-

Small-parsimony problem-

DP algorithm for solving small parsimony problem-

Large-parsimony problem-

nearest-neighbor interchange-

**Chapter 8:**

Diauxic shift-

Gene expression matrix-

Clustering as an optimization problem-

k-means clustering problem-

Lloyd algorithm-

Select k arbitrary starting nodes as “centers” and organize nodes into clusters based on their distance to the nearest center. From within each cluster, recalculate which node is closest to the cluster center and select it as the new center node. Recalculate clusters based on location to nearest center node. Repeat process.

coin flipping problem-

EM algorithm-

soft k-means clustering-

Distance-based clustering problem-

Hierarchical clustering-