Each question carries 5 points
Attempt all questions
Plagiarism is punishable
Time 80 minutes

- **Show the steps of searching GT in CGTAACGGT, using BWT.**

Reverse, indexing and querying

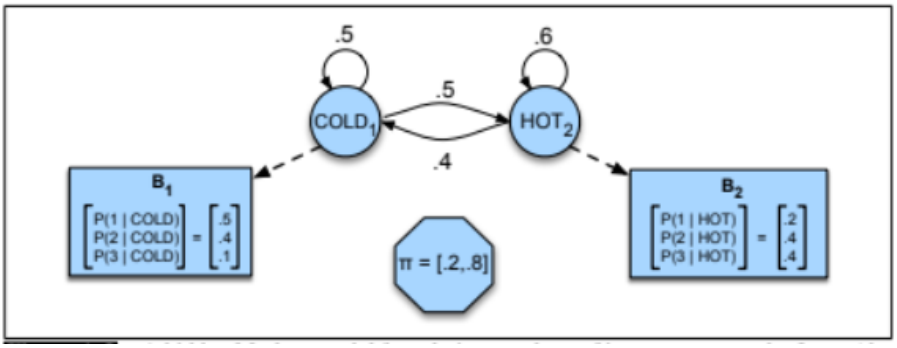
Querying

- **Define HMM, and its different components. What are some of its applications? Provide adequate explanations. (1+2+1+1)**

A hidden Markov model (HMM) allows us to talk about both observed events Hidden Markov model (like words that we see in the input) and qwhidden events (like part-of-speech tags) that we think of as causal factors in our probabilistic model. An HMM is specified by the following components:

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, ..., v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, ..., \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

Figure shows a sample HMM for the ice cream task. The two hidden states (H and C) correspond to hot and cold weather, and the observations (drawn from the alphabet O = {1,2,3}) correspond to the number of ice creams eaten by Jason on a given day.



❖ **What is homoplasy? Write the pseudo code for Fitch algorithm for Maximum Parsimony. (2 + 3)**

A homoplasy is the **opposite of a homology**, where a common ancestor provided the genes that gave rise to the trait in two or more animals. Often, a homoplasy will occur when two very different groups of animals evolve to do the same thing. This is known as convergent evolution, or convergence.

Algorithm for determining length of given tree: Fitch

- Root the tree at an arbitrary internal node (or internal branch)
- Visit an internal node x for which no state set has been defined, but where the state sets of x's immediate descendants (y,z) have been defined.
- If the state sets of y,z have common states, then assign these to x.
- If there are no common states, then assign the union of y,z to x, and increase tree length by one.
- Repeat until all internal nodes have been visited. Note length of current tree. ■

❖ **What is Locality Sensitive Hashing? Explain how it speeds up FiRE, considering an exhaustive nearest neighbor search as benchmark.**

Locality Sensitive Hashing (LSH) is a technique for efficiently finding approximate

nearest neighbors in high-dimensional spaces. It works by using hash functions to map high-dimensional data points to a lower-dimensional space in such a way that points that are close to each other in the high-dimensional space are likely to be mapped to the same location in the lower-dimensional space.

LSH speeds up nearest neighbor search by reducing the number of points that need to be considered during the search. Rather than exhaustively searching through all points in the dataset, LSH allows you to search only through a subset of points that are likely to be close to the query point. This is done by partitioning the points into "buckets" based on their hash values and only considering points in the same bucket as the query point.

In the context of FiRE (Fast Randomized Euclidean distance), LSH can be used to speed up the computation of approximate nearest neighbors. FiRE is an algorithm for efficiently computing approximate nearest neighbors in Euclidean space using random projections. By combining LSH with FiRE, you can further reduce the number of points that need to be considered during the search, leading to even faster search times.

❖ **Take two random phylogeny trees of your choice and calculate tree lengths given the following:**

	1	2	3	4	5	6	7
Human	c	c	t	t	g	a	a
Chimp	c	c	t	t	g	a	a
Gorilla	c	c	t	a	g	t	a
Gibbon	t	c	a	a	g	a	a
Orangutan	t	c	a	a	g	a	t

To calculate the tree lengths of these two trees, we need to consider the lengths of the branches connecting the nodes and the number of substitutions that occurred on each branch. Let's assume that all branches have a length of 1 unit.

For Tree 1, the tree length is calculated as follows:

Branch connecting A and B: 1 unit, 3 substitutions (ccttgaa -> cctagta)
Branch connecting A and C: 1 unit, 2 substitutions (ccttgaa -> tcaagaa)
Branch connecting B and D: 1 unit, 2 substitutions (cctagta -> tcaagaa)
Branch connecting C and E: 1 unit, 1 substitution (tcaagaa -> tcaagat)
Branch connecting C and F: 1 unit, 0 substitutions (tcaagaa -> tcaagaa)

The total tree length for Tree 1 is therefore 5 units and 6 substitutions.

For Tree 2, the tree length is calculated as follows:

Branch connecting A and B: 1 unit, 3 substitutions (ccttgaa -> cctagta)
Branch connecting A and C: 1 unit, 2 substitutions (ccttgaa -> tcaagaa)
Branch connecting B and D: 1 unit, 2 substitutions (cctagta -> tcaagaa)
Branch connecting C and E: 1 unit, 1 substitution (tcaagaa -> tcaagat)
Branch connecting C and F: 1 unit, 0 substitutions (tcaagaa -> tcaagaa)
Branch connecting E and G: 1 unit, 1 substitution (tcaagat -> tcaagaa)
Branch connecting E and H: 1 unit, 0 substitutions (tcaagat -> tcaagat)
Branch connecting F and I: 1 unit, 0 substitutions (tcaagaa -> tcaagaa)
The total tree length for Tree 2 is therefore 8 units and 7 substitutions.