Data Structures and Algorithms
INFO 6205
Homework 8
Due: Sunday 24, 2019

Put all your java, compiled class files and documentation files into a zip file named Homework8.zip and submit it via the drop box on the blackboard before the END of due date. Put your name on all .java files. There will be a short Quiz on this homework.

1. A MinHeap is a complete binary tree where the minimum-valued element is stored at the root node and where every node is less than or equal to both of its child nodes.

Note: Java code for MaxHeap algorithm is provided in attached file.

a) Compile and test MaxHeap code.

b) Modify MaxHeap code to MinHeap, and then compile and test the code.

2. Consider the following Text and Pattern

Text: ABCADCBABABCDABCDABDE
Pattern: BCD

a) Apply Brute-Force substring search algorithm to scan pattern in the following text string. Show step-by-step of the algorithm. Write Java code for the algorithm for input data. What is time complexity?

T	**A**	**B**	**C**	**A**	**D**	**C**	**B**	**A**	**B**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**	**A**	**B**	**D**	**E**
P	B	C	D																		
		B	C	D																	
			B	C	D																
				B	C	D															
					B	C	D														
						B	C	D													
							B	C	D												
								B	C	D											
									B	C	D										
										B	C	D									
											B	C	D								
												B	C	D							
													B	C	D						
														B	C	D					
															B	C	D				
																B	C	D			
																	B	C	D		
																		B	C	D	
																			B	C	D

The time complexity is O(n*m).

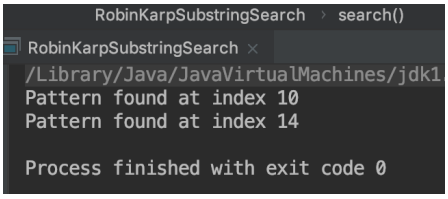
b) Apply Robin-Karp substring search algorithm to scan pattern in the following text string. Show step-by-step of the algorithm. Write Java code for the algorithm for input data. What is time complexity?

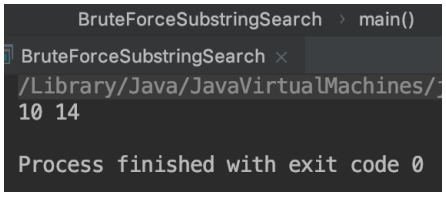
Hashcode of BCD is 56.

T	Hashcode compare with BCD	A	B	C	A	D	C	B	A	B	A	B	C	D	A	B	C	D	A	B	D	E
P	ABC	B	C	D																		
	BCA		B	C	D																	
	CAD			B	C	D																
	ADC				B	C	D															
	DCB					B	C	D														
	CBA						B	C	D													
	BAB							B	C	D												
	ABA								B	C	D											
	BAB									B	C	D										
	ABC										B	C	D									
	BCD											B	C	D								
	CDA												B	C	D							
	DAB													B	C	D						
	ABC														B	C	D					
	BCD															B	C	D				
	CDA																B	C	D			
	DAB																	B	C	D		
	ABD																		B	C	D	
	BDE																			B	C	D

The time complexity is O(n*m).

c) The time complexity of both algorithms in (a) and (b) is O(mn). What is the difference between the two time complexity? Show Java output graphically.





3. Consider Knuth-Morris-Paratt substring search Algorithm:
https://www.ics.uci.edu/~eppstein/161/960227.html

a) How the algorithm works for Text="bananobano", and Pattern="nano"

T	b	a	n	a	n	a	n	o	b	a	n	o
P	n	a	n	o								
		n	a	n	o							
			n	a	n	o						
					n	a	n	o				
								n	a	n	o	
									n	a	n	o

b) How Knuth-Morris-Paratt is different from Brute-force algorithm?

The KMP algorithm saves more time and is more efficient when searching. In the brute force algorithm, when pattern and text do not match, it is necessary to move the matches one by one in order, resulting in a time complexity of O(n*m). However, when there is a mismatch in KMP, it will avoid repeat matching again, and will directly move the pattern to the previously matched part, and match the next character. The time complexity of KMP is O(n).

4. Read the following reference:
https://www.hackerearth.com/practice/algorithms/greedy/basics-of-greedy-algorithms/tutorial/

a) Solve the greedy problem for A={8,7,6,5,4,3,2,1}, T=15

After the 1st iteration:	currentTime = 1 numberOfThings = 1
After the 2nd iteration	currentTime is 1 + 2 = 3 numberOfThings = 2
After the 3rd iteration:	currentTime is 3 + 3 = 6 numberOfThings = 3
After the 4th iteration:	currentTime is 6 + 4 = 10 numberOfThings = 4
After the 5th iteration:	currentTime is 10 + 5 = 15 numberOfThings = 5

After the 5th iteration, currentTime is15, which is equal to T. Therefore, the answer is 5.

b) Write Java code for greedy algorithm with input data in (a)

c) Consider the Scheduling program, how does that work?

Objective function F is the weighted sum of the completion times.
F = P[1] * C(1) + P[2] * C(2) + ... + P[N] * C(N)
This objective function must be minimized.

If the time required to complete a different task is the same, ie T [i] = T [j], where 1 <= i,j <= N, but they have different priorities. In order to make the objective function as small as possible, the highest priority must be associated with the shortest completion time.

If different tasks have the same priority, ie P [i] = P [j], where 1 <= i,j <= N but their lengths are different. Give priority to the following tasks: have higher priority and take less time to complete.

In general:
Prioritize higher priority so that higher priority leads to higher scores.
Prioritize tasks that take less time to complete so that the longer the time re-quired, the lower the score.

d) How (c) is different in (a)?

The scheduling problem is a two-dimensional problem that requires time and weight to be considered in order to calculate a reasonable result. Relative to a), a) is easier, he only considers the arrangement of time.

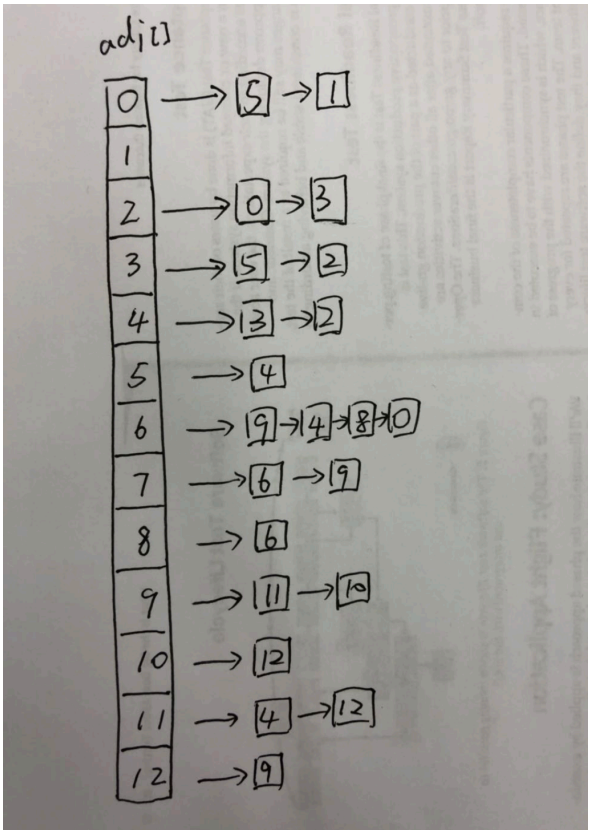
5. a) Consider Graph data structures for Adjancey-Lists for both directed graph and undirected Graph

b) What are the differences (data structures, methods)

	Directed graph	Undirected Graph
Data structures	Vertices are stored in a one-dimensional array. In the vertex array, each data element also needs to store a pointer to the first neighbor.	Vertices are stored in a one-dimensional array. In the vertex array, each data element also needs to store a pointer to the first neighbor.
Methods	It is easy to get the degree of each vertex by using the vertex as the end of the arc to store the edge table, and the table of the vertex as the head of the arc is easy to get the inversion of the vertex, that is, the inverse adjacency list.	All the adjacent points of each vertex constitute a linear table. Because the number of adjacent points is indefinite, it is stored in a single linked list. The undirected graph is called the edge table of the vertex vi, and the directed graph is called the vertex as the outbound table of the arc tail.

6. Consider the following directed graph:

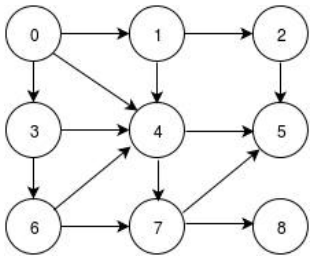
a) Construct Adjancey-List for Graph



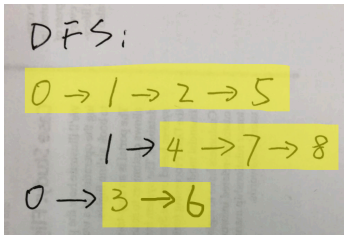

```
DiGraph
/Library/Java/JavaVirtualMach
Adjancey List is :
13 vertexes, 22 edges
0: [1, 5]
1: []
2: [0, 3]
3: [5, 2]
4: [2, 3]
5: [4]
6: [0, 4, 8, 9]
7: [6, 9]
8: [6]
9: [11, 10]
10: [12]
11: [12, 4]
12: [9]
```

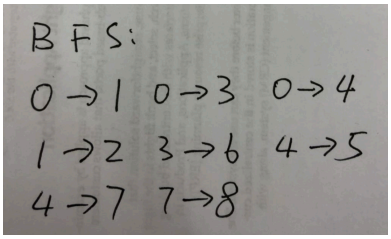
b) Write Java code to iterate through Graph and print the connection for each Vertex.

7. Consider the following Directed Graph:



A) That is the DFS and BFS on this graph, show step-by-step





B) What is time complexity for each case?
The time complexity for both is O(V+E).

C) Write Java code, what is the termination for recursive calls
It terminates when the traversal is terminated when it is traversed to the same node.

Lecture8Notes

Greedy Algorithm
https://www.tutorialspoint.com/data_structures_algorithms/greedy_algorithms.htm
Substring Search
https://www.globalsoftwaresupport.com/brute-force-substring-search/
https://brilliant.org/wiki/rabin-karp-algorithm/
https://algs4.cs.princeton.edu/lectures/53SubstringSearch.pdf
See slides for directed graph and undirected