

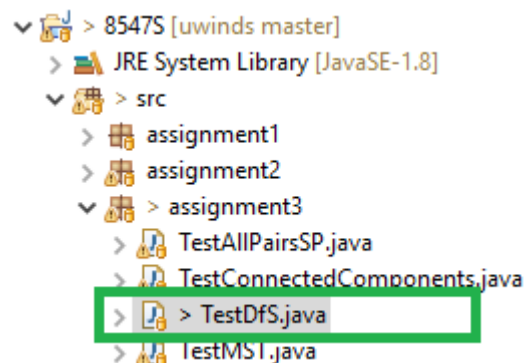
Assignment 3

As a student at the University of Windsor, I pledge to pursue all endeavours with honour and integrity and will not tolerate or engage in academic or personal dishonesty. I confirm that I have not received any unauthorized assistance in preparing for or writing this assignment. I acknowledge that a mark of 0 may be assigned for copied work.

Wen Dong #110057395

Task #1

1. Source code



2. Output of the java file looks as below:

Task #1 - Run DFS on mediumDG.txt. Time spent:

Creating DG from mediumDG.txt - 178 ms

DFS - 6 ms

Pre-order: 0 34 40 15 49 22 6 28 44 39 42 31 32 47 30 37 35 46 16 9 11 21 5

43 38 23 13 18 29 8 48 33 19 2 25 12 26 7 41 1 45 14 3 36 20 27 24 4 17 10

Post-order: 21 11 9 16 46 35 38 43 5 37 23 30 47 13 32 18 31 42 39 44 29 28 6

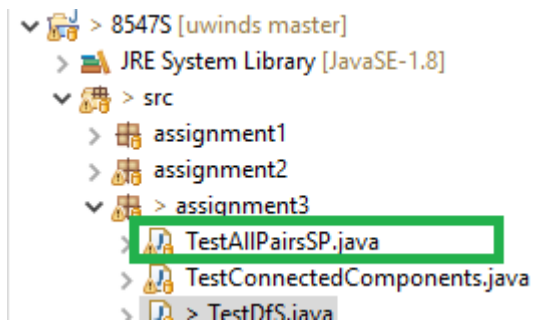
22 49 19 25 2 33 26 12 48 8 15 40 34 41 7 0 14 45 1 20 36 17 4 24 27 3 10

3. The CPU time for DFS is 6 ms as per the output.

4. The Worst-case complexity of DFS is Running time of DFS is $O(|V|+|E|)$, if it happens to be the last node visited as the worst case, it will need to traverse the entire graph, i.e. traversing all vertexes and edges which is of time complexity $O(|V|+|E|)$

Task #2 - a

1. Source code



2. Output of the java file looks as below:

Task #2 - Find shortest path for all pairs in mediumEWG.txt
Total time spent: 126 ms.

Here is a snippet of Shortest paths matrix (10 X 10), you can run it and see full results in resulted "sp_all_pairs.txt"

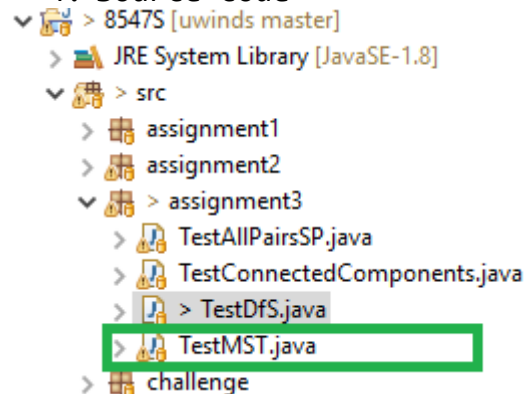
0.00	N	N	N	N	N	N	N	N	N
N	0.00	N	N	N	N	N	N	N	N
N	N	0.00	N	N	N	N	N	N	N
N	N	N	0.00	N	N	N	N	N	N
N	N	N	N	0.00	0.11	N	N	N	N
N	N	N	N	N	0.00	N	N	N	N
N	N	N	N	N	N	0.00	N	N	N
N	N	N	N	N	N	N	0.00	N	N
N	N	N	N	N	N	N	N	0.00	N
N	N	N	N	N	N	N	N	N	0.00

3. Report entire time complexity

The time complexity of my algorithm is $O(E^2 + |E||V| \log |V|)$, as it apply Dijkstra search to each vertex as source to calculate shortest paths to all the rest. As we learned from class, Dijkstra's time complexity is $O(E + |V| \log |V|)$, so the entire cost is $O(E^2 + |E||V| \log |V|)$.

Task #2 - b

4. Source code



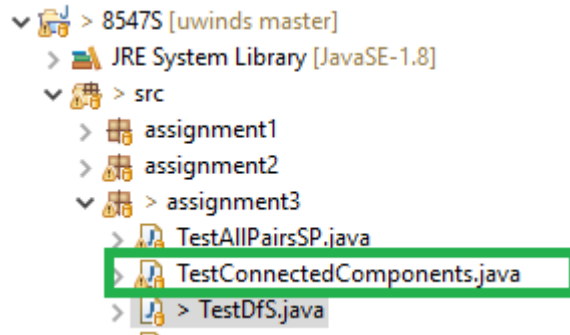
5. Output of the java file looks as below:

```
Finds MST in mediumEWG.txt, time spent:
loading EWG from mediumEWG.txt: 136 ms
finding MST: 17 ms
```

6. The algorithm in my code is Kruskal which is of time complexity (worst-case) $O(|E| \log |E|)$.

Task #3 and #4

1. Source code



2. Output of the java file looks as below:

33 connected components found in movies.txt

Task #3 - It took 49 ms to use DFS identifying the components and 33 connected components identified in movies.txt.

Task #4 -

Movies starred by DiCaprio, Leonardo:

What's Eating Gilbert Grape (1993)
Total Eclipse (1995)
Titanic (1997)
This Boy's Life (1993)
Romeo + Juliet (1996)
Quick and the Dead, The (1995)
Poison Ivy (1992)
Marvin's Room (1996)
Man in the Iron Mask, The (1998 I)
Gangs of New York (2002)
Departed, The (2006)
Celebrity (1998)
Catch Me If You Can (2002)
Beach, The (2000 I)
Basketball Diaries, The (1995)
Aviator, The (2004)

Movies starred by Roberts, Julia (I):

Stepmom (1998)
Steel Magnolias (1989)
Something to Talk About (1995)
Sleeping with the Enemy (1991)
Runaway Bride (1999)
Prêt-à-Porter (1994)
Pretty Woman (1990)
Player, The (1992)
Pelican Brief, The (1993)
Ocean's Twelve (2004)

Ocean's Eleven (2001)
 Notting Hill (1999)
 Mystic Pizza (1988)
 My Best Friend's Wedding (1997)
 Mona Lisa Smile (2003)
 Michael Collins (1996)
 Mexican, The (2001)
 Mary Reilly (1996)
 I Love Trouble (1994)
 Hook (1991)
 Full Frontal (2002)
 Flatliners (1990)
 Everyone Says I Love You (1996)
 Erin Brockovich (2000)
 Dying Young (1991)
 Conspiracy Theory (1997)
 Confessions of a Dangerous Mind (2002)
 Closer (2004 I)
 America's Sweethearts (2001)

Movies starred by Grant, Hugh (I):

Two Weeks Notice (2002)
 Small Time Crooks (2000)
 Sirens (1994)
 Sense and Sensibility (1995)
 Restoration (1995)
 Remains of the Day, The (1993)
 Notting Hill (1999)
 Nine Months (1995)
 Mickey Blue Eyes (1999)
 Maurice (1987)
 Love Actually (2003)
 Lair of the White Worm, The (1988)
 Four Weddings and a Funeral (1994)
 Extreme Measures (1996)
 Englishman Who Went Up a Hill But Came Down a Mountain, The (1995)
 Bridget Jones: The Edge of Reason (2004)
 Bridget Jones's Diary (2001)
 Bitter Moon (1992)
 American Dreamz (2006)
 About a Boy (2002)

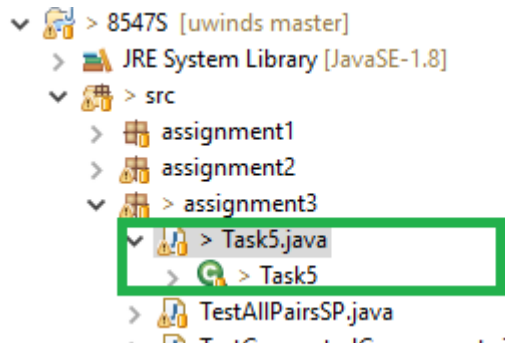
Movies starred by both Roberts, Julia (I) and Grant, Hugh (I):

Notting Hill (1999)

3. According to the output, the CPU time is 49 ms for DFS identifying the connected components, while the time complexity of the algorithm DFS which is in my implementation is $O(|V|+|E|)$ as we learned from the class. worst-case is the case $O(|V|+|E|)$ as well.
4. From the output, we can see the movies starred by the given actors respectively, and by both Roberts, Julia (I) and Grant, Hugh (I)

Task #5

1. Source code



2. Output of the java file looks as below:

Partitioning 'resources/ChIP-seq reads/ChIP-seq-reads-1M.dat' into following files:

output/A.dat

output/B.dat

output/C.dat

output/D.dat

Sorting output/A.dat as output/AS.dat.

Sorting output/B.dat as output/BS.dat.

Sorting output/C.dat as output/CS.dat.

Sorting output/D.dat as output/DS.dat.

Merge sorting following files:

output/AS.dat

output/BS.dat

output/CS.dat

output/DS.dat

Merge Sorting done into 'output/Chip-seq-reads-1M-sorted.dat'

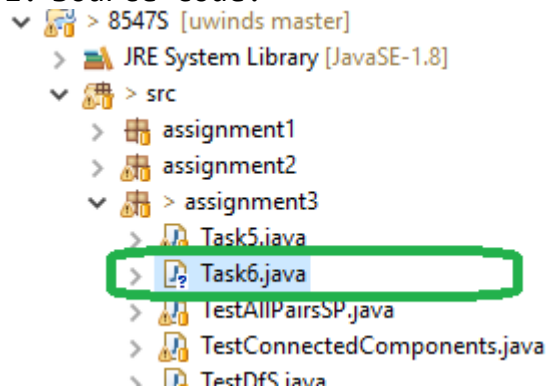
Total CPU time is 4052 ms

3. Time complexity comments:

It uses binary heap for sorting a single file sub file like A.dat, the time complexity is $O(n \log n)$ as we learned from class; On the other hand, it uses IndexMinPQ for multiway merge, the time complexity is $O(n \log d)$ according to what we learned from class that finding the minimum from the d-way IndexMinPQ is $O(\log d)$ and we have in all n elements to sort.

Task #6

1. Source code:



Apart from Task6.java, I have modified following file a bit to meet in-order traverse.

memoryManagement/ BTree.java

2. Output of the java file looks as below:

```
Loading into BTree - 'resources/ChIP-seq reads/ChIP-seq-reads-1M.dat'.
Write the keys (in al 1000000) in in-order to 'output/B-tree.dat'.
Done. Total CPU time is 2405 ms
```

3. Comment on the obtained CPU times:

As we learn from class, time complexity of search, insertion and removal for B-Tree is $O(\log_B n)$, while B is the order and N is the total number of elements. So, the insertions during building BTree from is $O(N \log_B N)$. In-order traverse is of $O(N)$.

Task #7 - done respectively inline for #5 and #6