

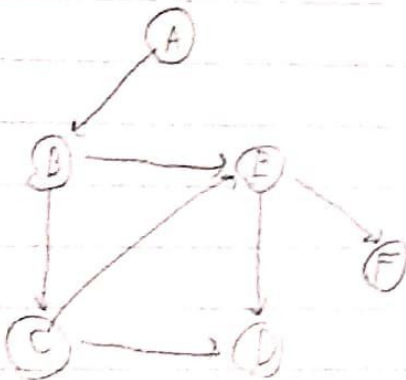
ឃ្លាត ៨ កាត

- 1 ឃ្លាតទិន្នន័យដើម
- 2 កាតដើមកំណត់ដំបូង
- 3 កាតដើមកំណត់ដើម
- 4 ទិន្នន័យកាតដើមកំណត់ដើម
- 5 ទិន្នន័យកាតដើមកំណត់ដើម
- 6 កាតដើម
- 7 កាតដើមកំណត់ដើម
- 8 កាតដើមកំណត់ដើម
- 9 កាតដើមកំណត់ដើម
- 10 កាតដើមកំណត់ដើម
- 11 កាតដើមកំណត់ដើម
- 12 កាតដើមកំណត់ដើម
- 13 កាតដើមកំណត់ដើម
- 14 កាតដើម
- 15 កាតដើមកំណត់ដើម
- 16 កាតដើមកំណត់ដើម
- 17 កាតដើមកំណត់ដើម
- 18 កាតដើម
- 19 កាតដើម
- 20 កាតដើមកំណត់ដើម
- 21 កាតដើមកំណត់ដើម
- 22 កាតដើមកំណត់ដើម
- 23 កាតដើមកំណត់ដើម Preorder
- 24 កាតដើមកំណត់ដើម
- 25 កាតដើមកំណត់ដើម
- 26 កាតដើមកំណត់ដើម
- 27 កាតដើមកំណត់ដើម
- 28 កាតដើមកំណត់ដើម
- 29 Minimum Spanning Tree
- 30 កាតដើមកំណត់ដើម

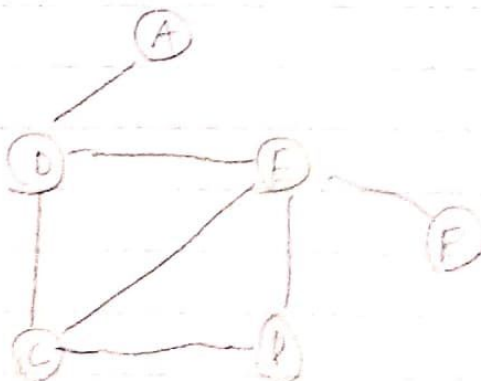
8

- 1 กราฟ (Graph) คือ กรอบโครงข่ายที่ประกอบด้วย จุดยอด (Vertex / Vertices) และ กรอบโครงข่าย (Edges) ซึ่งก็จะเป็นเส้นที่เชื่อมระหว่างจุดยอด กราฟที่ประกอบด้วยจุดยอดและขอบที่เชื่อมระหว่างจุดยอด กราฟที่ประกอบด้วยจุดยอดและขอบที่เชื่อมระหว่างจุดยอด กราฟที่ประกอบด้วยจุดยอดและขอบที่เชื่อมระหว่างจุดยอด
- 2 กราฟที่มีทิศทางเรียกว่า ไดกราฟ (Digraph) เป็นกราฟที่มีเส้นเชื่อมที่มีทิศทางไปข้างหน้า Successor โดยเริ่มต้นที่จุดยอดและจบลงที่จุดยอดในกราฟที่มีทิศทางเรียกว่า อาร์ค (Arcs)
- 3 กราฟที่ไม่มีทิศทางคือ กราฟที่ไม่ได้ระบุทิศทาง กราฟที่ไม่มีทิศทางเรียกว่า กราฟที่ไม่มีทิศทาง กราฟที่ไม่มีทิศทางเรียกว่า กราฟที่ไม่มีทิศทาง กราฟที่ไม่มีทิศทางเรียกว่า กราฟที่ไม่มีทิศทาง

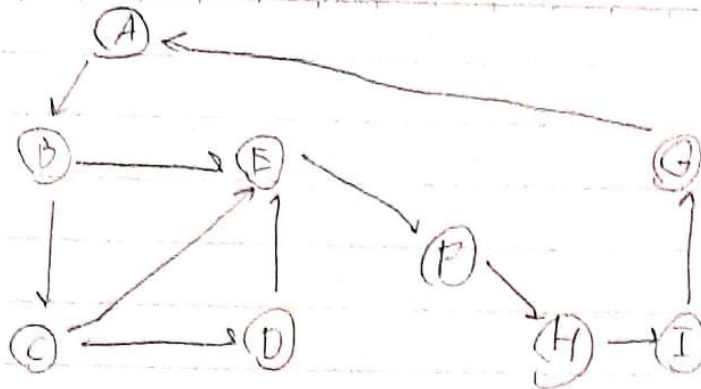
4



5



- 6 เส้นทาง (Path) คือ ลำดับของจุดยอดที่เชื่อมต่อกันโดยเส้นเชื่อมที่ต่อเนื่องกัน ตัวอย่างเช่น ลำดับของจุดยอด {A, B, C, D} เป็นเส้นทาง และ ลำดับของจุดยอด {A, B, E, F} ก็เป็นเส้นทาง



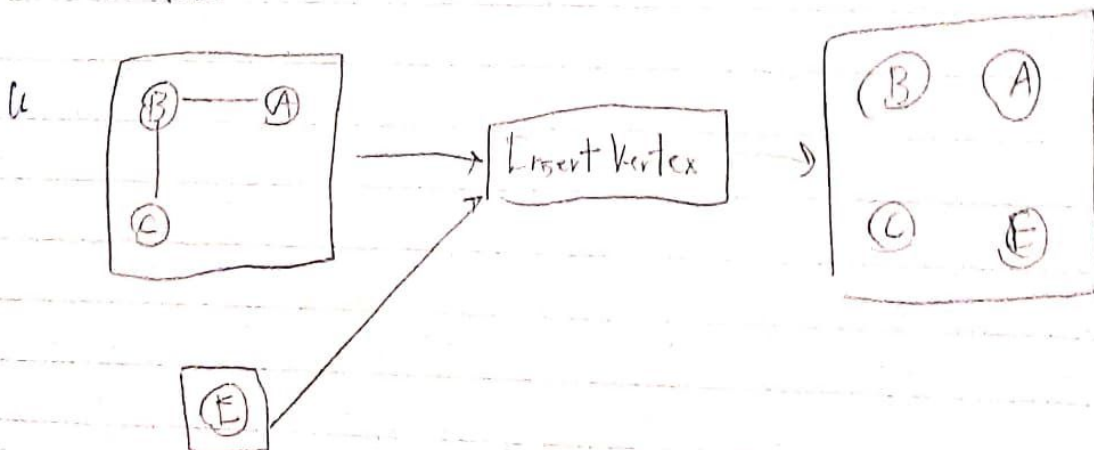
```

graph TD
    A((A)) --> B((B))
    B --> E((E))
    B --> C((C))
    C --> D((D))
    D --> E
    E --> F((F))
    H((H)) --> G((G))

```

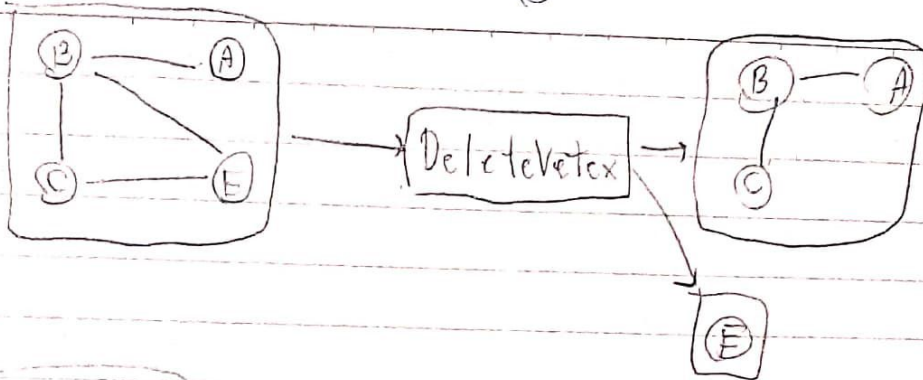
14. อัตราส่วนของพื้นที่สอง จำนวนสาม (โดยนัยของอาร์ด) ที่เชื่อมโยงไว้กับ วงรีที่
จำนวนอัตราส่วนของพื้นที่ B คือ 3 และอัตราส่วนของพื้นที่ E คือ 4 แสดงว่า
อัตราส่วนของพื้นที่สองคือ 1:2:3:4:5:6:7:8:9:10:11:12:13:14:15:16:17:18:19:20:21:22:23:24:25:26:27:28:29:30:31:32:33:34:35:36:37:38:39:40:41:42:43:44:45:46:47:48:49:50:51:52:53:54:55:56:57:58:59:60:61:62:63:64:65:66:67:68:69:70:71:72:73:74:75:76:77:78:79:80:81:82:83:84:85:86:87:88:89:90:91:92:93:94:95:96:97:98:99:100:101:102:103:104:105:106:107:108:109:110:111:112:113:114:115:116:117:118:119:120:121:122:123:124:125:126:127:128:129:130:131:132:133:134:135:136:137:138:139:140:141:142:143:144:145:146:147:148:149:150:151:152:153:154:155:156:157:158:159:160:161:162:163:164:165:166:167:168:169:170:171:172:173:174:175:176:177:178:179:180:181:182:183:184:185:186:187:188:189:190:191:192:193:194:195:196:197:198:199:200:201:202:203:204:205:206:207:208:209:210:211:212:213:214:215:216:217:218:219:220:221:222:223:224:225:226:227:228:229:230:231:232:233:234:235:236:237:238:239:240:241:242:243:244:245:246:247:248:249:250:251:252:253:254:255:256:257:258:259:260:261:262:263:264:265:266:267:268:269:270:271:272:273:274:275:276:277:278:279:280:281:282:283:284:285:286:287:288:289:290:291:292:293:294:295:296:297:298:299:300:301:302:303:304:305:306:307:308:309:310:311:312:313:314:315:316:317:318:319:320:321:322:323:324:325:326:327:328:329:330:331:332:333:334:335:336:337:338:339:340:341:342:343:344:345:346:347:348:349:350:351:352:353:354:355:356:357:358:359:360:361:362:363:364:365:366:367:368:369:370:371:372:373:374:375:376:377:378:379:380:381:382:383:384:385:386:387:388:389:390:391:392:393:394:395:396:397:398:399:400:401:402:403:404:405:406:407:408:409:410:411:412:413:414:415:416:417:418:419:420:421:422:423:424:425:426:427:428:429:430:431:432:433:434:435:436:437:438:439:440:441:442:443:444:445:446:447:448:449:450:451:452:453:454:455:456:457:458:459:460:461:462:463:464:465:466:467:468:469:470:471:472:473:474:475:476:477:478:479:480:481:482:483:484:485:486:487:488:489:490:491:492:493:494:495:496:497:498:499:500:501:502:503:504:505:506:507:508:509:510:511:512:513:514:515:516:517:518:519:520:521:522:523:524:525:526:527:528:529:530:531:532:533:534:535:536:537:538:539:540:541:542:543:544:545:546:547:548:549:550:551:552:553:554:555:556:557:558:559:560:561:562:563:564:565:566:567:568:569:570:571:572:573:574:575:576:577:578:579:580:581:582:583:584:585:586:587:588:589:590:591:592:593:594:595:596:597:598:599:600:601:602:603:604:605:606:607:608:609:610:611:612:613:614:615:616:617:618:619:620:621:622:623:624:625:626:627:628:629:630:631:632:633:634:635:636:637:638:639:640:641:642:643:644:645:646:647:648:649:650:651:652:653:654:655:656:657:658:659:660:661:662:663:664:665:666:667:668:669:670:671:672:673:674:675:676:677:678:679:680:681:682:683:684:685:686:687:688:689:690:691:692:693:694:695:696:697:698:699:700:701:702:703:704:705:706:707:708:709:710:711:712:713:714:715:716:717:718:719:720:721:722:723:724:725:726:727:728:729:730:731:732:733:734:735:736:737:738:739:740:741:742:743:744:745:746:747:748:749:750:751:752:753:754:755:756:757:758:759:760:761:762:763:764:765:766:767:768:769:770:771:772:773:774:775:776:777:778:779:780:781:782:783:784:785:786:787:788:789:790:791:792:793:794:795:796:797:798:799:800:801:802:803:804:805:806:807:808:809:810:811:812:813:814:815:816:817:818:819:820:821:822:823:824:825:826:827:828:829:830:831:832:833:834:835:836:837:838:839:840:841:842:843:844:845:846:847:848:849:850:851:852:853:854:855:856:857:858:859:860:861:862:863:864:865:866:867:868:869:870:871:872:873:874:875:876:877:878:879:880:881:882:883:884:885:886:887:888:889:890:891:892:893:894:895:896:897:898:899:900:901:902:903:904:905:906:907:908:909:910:911:912:913:914:915:916:917:918:919:920:921:922:923:924:925:926:927:928:929:930:931:932:933:934:935:936:937:938:939:940:941:942:943:944:945:946:947:948:949:950:951:952:953:954:955:956:957:958:959:960:961:962:963:964:965:966:967:968:969:970:971:972:973:974:975:976:977:978:979:980:981:982:983:984:985:986:987:988:989:990:991:992:993:994:995:996:997:998:999:1000:1001:1002:1003:1004:1005:1006:1007:1008:1009:1010:1011:1012:1013:1014:1015:1016:1017:1018:1019:1020:1021:1022:1023:1024:1025:1026:1027:1028:1029:10

- 15 การกำหนดขององค์กร คือ การแบ่งงานกันทำโดยมีขอบเขตที่แน่นอน ในการทำงาน
ตามวัตถุประสงค์ขององค์กรไปทั่วทุกส่วนงาน องค์กรที่มีจะยังไม่ถึง 100 คน ให้ใช้ 1 คน
ต่อ 1 งาน แต่ถ้ามี 100 คนขึ้นไป องค์กรที่มากกว่านั้นจะให้มีคนส่วนน้อยเท่ากับองค์กรใน
ในบริษัท

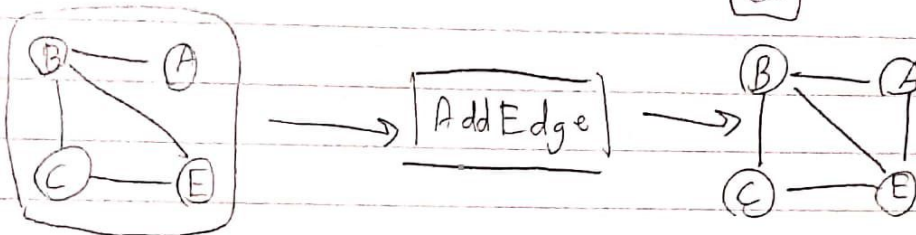


(6)

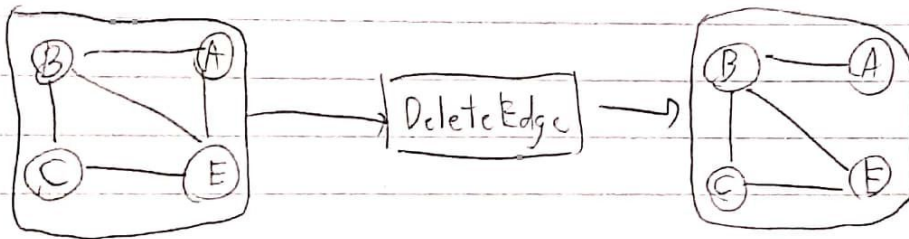
17



18



19

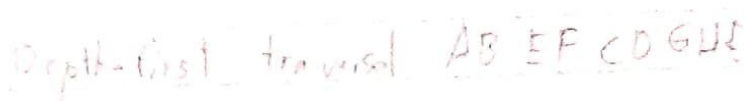


20

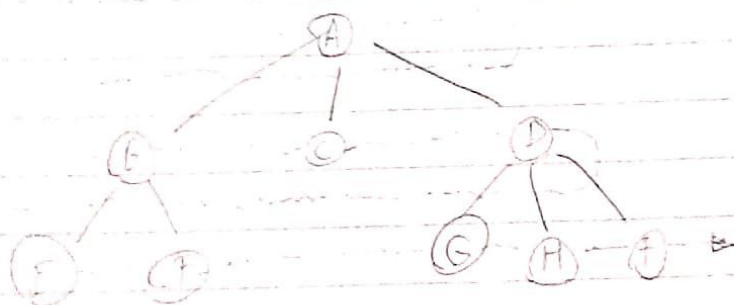


21 การหาเส้นทางในกราฟ โดยหาว่าเส้นทางที่สั้นที่สุดคือเส้นทางใด และหาว่าเส้นทางที่สั้นที่สุดมีกี่เส้นทาง

22 เป็นวิธีที่หาเส้นทางในกราฟที่มีน้ำหนักทุกจุด และหาว่าเส้นทางที่สั้นที่สุดมีกี่เส้นทาง และหาว่าเส้นทางที่สั้นที่สุดมีกี่เส้นทาง



- 24) ဒီဇင်ဘာလ ၂၀၁၈ ခုနှစ် (အောက်)
 သက်တမ်း ၈၀ နှစ်အထိ
 အောက်ပါအတိုင်း

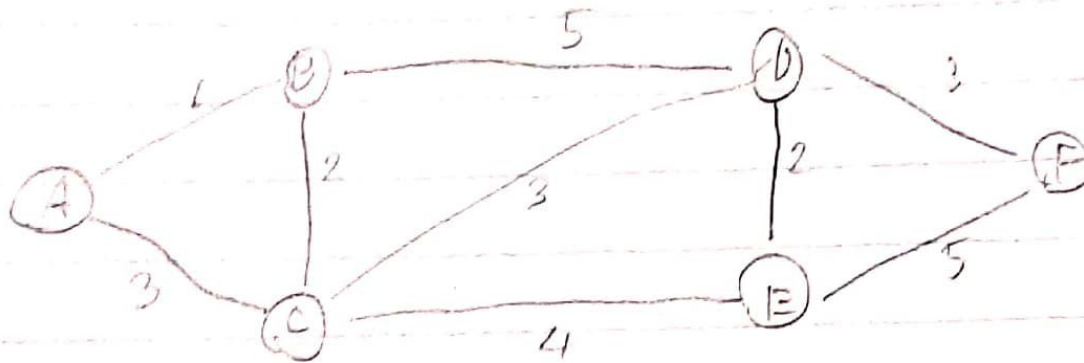


Breadth-first traversal ABCDEF GHI

- [illegible]

(F6)

- [illegible]



Unit 4 Sorting

- 1 Insertion Sort
- 2 Bubble Sort
- 3 Quick Sort
- 4 Selection Sort
- 5 Merge Sort
- 6 Heap Sort
- 7 Radix Sort
- 8 Selection Sort
- 9 Heap Sort
- 10 Insertion Sort
- 11 Bubble Sort
- 12 Quick Sort
- 13 Merge Sort
- 14 Selection Sort
- 15 Selection Sort
- 16 Selection Sort
- 17 Heap Sort
- 18 Selection Sort
- 19 Insertion Sort
- 20 Bubble Sort
- 21 Merge Sort
- 22 Quick Sort
- 23 Merge Sort
- 24 Selection Sort
- 25 Selection Sort
- 26 Insertion Sort
- 27 Bubble Sort
- 28 Median Sort
- 29 Merge Sort



1. Internal Sorting គឺជាប្រភេទនៃការប្រើប្រាស់ការប្រើប្រាស់ទិន្នន័យដែលស្ថិតនៅក្នុងអង្គបញ្ជីតែមួយ។ វាមានប្រភេទជាច្រើនដូចជា Insertion Sort, Shell Sort, Selection Sort ។
2. External Sorting គឺជាប្រភេទនៃការប្រើប្រាស់ការប្រើប្រាស់ទិន្នន័យដែលស្ថិតនៅក្នុងអង្គបញ្ជីខាងក្រៅ។ វាមានប្រភេទជាច្រើនដូចជា Merge Sort, Heap Sort, Quick Sort ។
3. Sort Order គឺជាប្រភេទនៃការប្រើប្រាស់ការប្រើប្រាស់ទិន្នន័យដែលស្ថិតនៅក្នុងអង្គបញ្ជីតែមួយ។ វាមានប្រភេទជាច្រើនដូចជា Ascending, Descending ។
4. Sort Stability គឺជាប្រភេទនៃការប្រើប្រាស់ការប្រើប្រាស់ទិន្នន័យដែលស្ថិតនៅក្នុងអង្គបញ្ជីតែមួយ។ វាមានប្រភេទជាច្រើនដូចជា Stable, Unstable ។
5. Sort Complexity គឺជាប្រភេទនៃការប្រើប្រាស់ការប្រើប្រាស់ទិន្នន័យដែលស្ថិតនៅក្នុងអង្គបញ្ជីតែមួយ។ វាមានប្រភេទជាច្រើនដូចជា Time Complexity, Space Complexity ។
6. Sort Algorithm គឺជាប្រភេទនៃការប្រើប្រាស់ការប្រើប្រាស់ទិន្នន័យដែលស្ថិតនៅក្នុងអង្គបញ្ជីតែមួយ។ វាមានប្រភេទជាច្រើនដូចជា Bubble Sort, Insertion Sort, Selection Sort, Merge Sort, Quick Sort ។

(9)

8. เป็นอัลกอริทึมที่ง่ายที่สุดในการเรียงลำดับข้อมูล โดยจะเลือกค่าที่น้อยที่สุดมาวางไว้ที่ตำแหน่งแรก และทำซ้ำจนกว่าข้อมูลจะเรียงลำดับเรียบร้อย

9. ขั้นตอนการทำงานของ Selection Sort จะเลือกค่าที่น้อยที่สุดมาวางไว้ที่ตำแหน่งแรก และทำซ้ำจนกว่าข้อมูลจะเรียงลำดับเรียบร้อย

10. Insertion Sort จะนำข้อมูลมาใส่ในตำแหน่งที่เหมาะสม และทำซ้ำจนกว่าข้อมูลจะเรียงลำดับเรียบร้อย

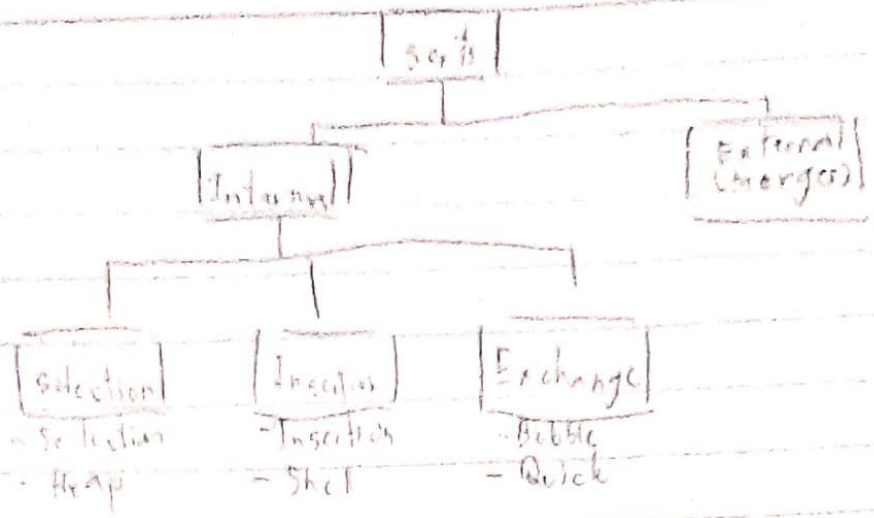
11. ขั้นตอนการทำงานของ Merge Sort จะนำข้อมูลมาแบ่งเป็นครึ่งๆ และทำซ้ำจนกว่าข้อมูลจะเรียงลำดับเรียบร้อย

12. ขั้นตอนการทำงานของ Bubble Sort และ Quick Sort จะนำข้อมูลมาเปรียบเทียบและแลกเปลี่ยนตำแหน่ง และทำซ้ำจนกว่าข้อมูลจะเรียงลำดับเรียบร้อย

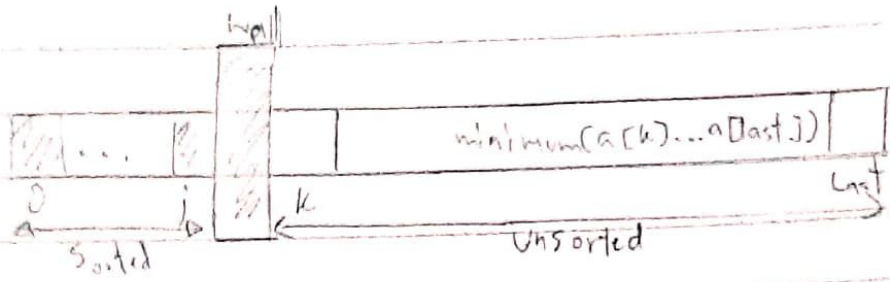
13. ขั้นตอนการทำงานของ File #1 และ File #2 จะนำข้อมูลมาเปรียบเทียบและแลกเปลี่ยนตำแหน่ง และทำซ้ำจนกว่าข้อมูลจะเรียงลำดับเรียบร้อย

14

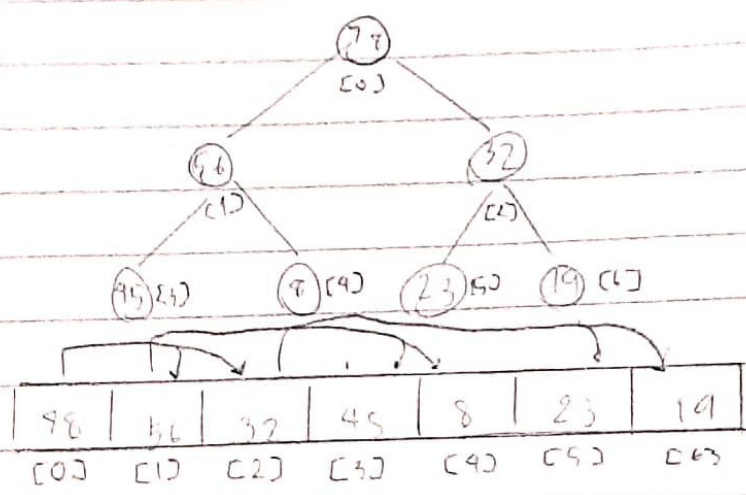
9



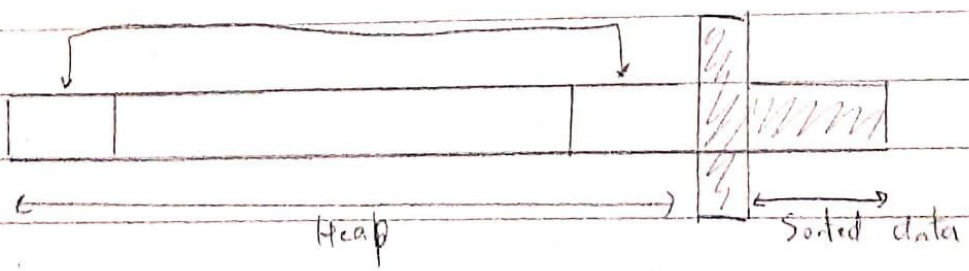
15



16

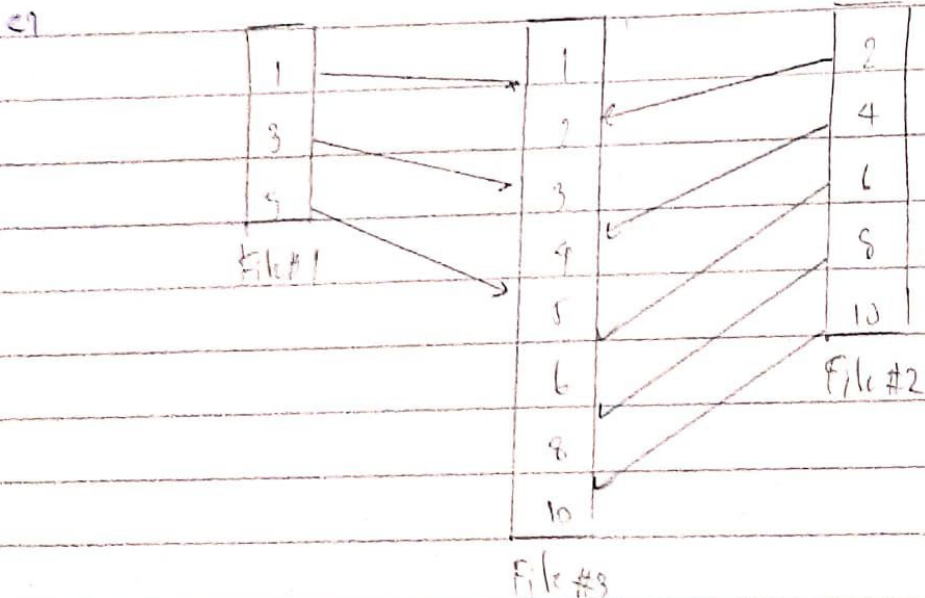
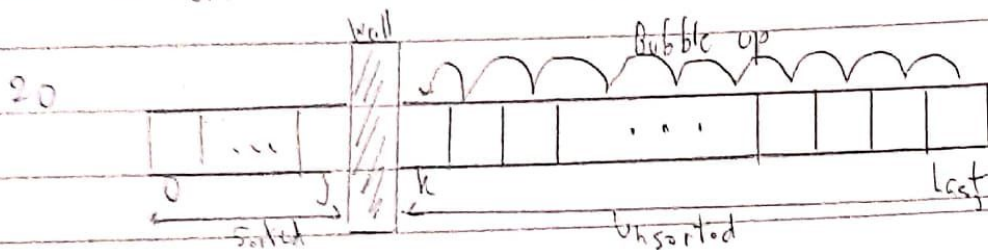
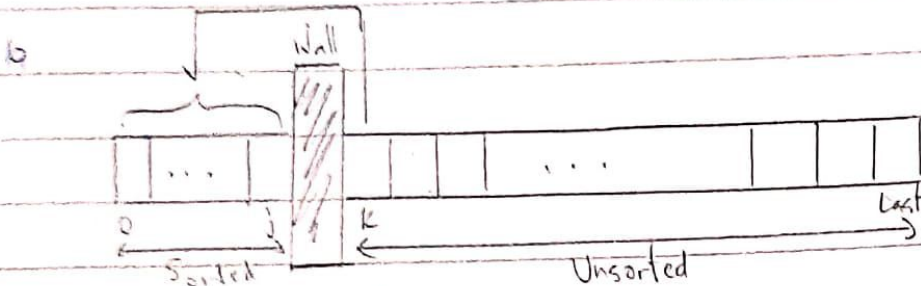


17



⑦

16	17	Number of loops	
		Selection Sort	Heap Sort
	25	625	112
	100	10,000	664
	500	250,000	4,482
	1000	1,000,000	9,965
	2000	4,000,000	10,965



(9)

22

Input File

2300 records

Sort

Records 1-500

1001-1500

2001-2300

Merge 1

501-1000

1501-2000

Merge 2

23

1 5 9 14 17 6 7 13

1 5 9 14

17 6 7 13

1 5 9 14

17 6 7 13

1 5 9 14

17 6 7 13

1 5 9 14

6 17 7 13

1 3 5 14

6 7 13 17

1 3 5 6 7 13 14 17

(9)

Q. No.	Sorting	Best case (Best-case)	Average case (Average case)	Worst case (Worst-case)
	Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
	Insertion Sort	$O(n)$	$O(n^2)$	$O(n^2)$
	Bubble Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$
	Quick Sort	$O(n \log n)$	$O(n \log n)$	$O(n^2)$
	Merge Sort	$O(n \log n)$	$O(n \log n)$	$O(n \log n)$

Q. Algorithm selection Sort (list, last)

1 set current to 0

2 loop (until last element sorted)

1 set smallest to current

2 set walker to current + 1

3 loop (walker <= last)

1 if (walker key < smallest key)

1 set smallest to walker

2 increment to walker

4 end loop

5 exchange (current, smallest)

6 increment current

3 end loop

end selectionSort

26 Algorithm heapSort (heap, last)

1 set walker to 1

2 loop (heap, walker)

1 reheapify (heap, walker)

2 increment walker

3 end loop

4 set sorted to last

5 loop until all data sorted

1 exchange heap 0, sorted

2 decrement sorted

3 reheapDown (heap, 0, sorted)

4 end loop

end heapSort

27 Algorithm insertionSort (list, last)

1 set current to 1

2 loop until last element sorted

1 move current element to hold

2 set walker to current - 1

3 loop (walker > 0 AND hold key < walker key)

1 move walker element right one element

2 decrement walker

4 end loop

5 move hold to walker + 1 element

6 increment current

7 end loop

end insertionSort

28 Algorithm bubbleSort (list, last)

1 set current to 0

2 set sorted to false

3 loop (current < last AND sorted false)

1 set walker to last

2 set sorted to true

3 loop (walker > current)

1 if (walker data < walker - 1 data)

1 set sorted to false

2 exchange (list, walker, walker - 1)

2 end if

3 decrement walker

4 end loop

5 increment current

6 end loop

end bubbleSort

29 Algorithm medianLeft (SortData, left, right)

1 set mid to $(\text{left} + \text{right}) / 2$

2 if (left key > mid key)

1 exchange (SortData, left, mid)

3 end if

4 if (left key > right key)

1 exchange (SortData, left, right)

5 end if

6 if (mid key > right key)

1 exchange (SortData, mid, right)

7 end if

8 exchange (SortData, left, mid)

end medianLeft

(9)

2.0 Algorithm mergeFiles

1 open files

2 read (file1 into record1)

3 read (file2 into record2)

4 loop (not end file1 OR not end file2)

1 if (record1 key < record2 key)

1 write (record1 to file3)

2 read (file1 into record1)

3 if (end of file1)

1 set record1 key to infinity

4 end if

2 else

1 write (record2 to file3)

2 read (file2 into record2)

3 if (end of file2)

1 set record2 key to infinity

4 end if

3 end if

5 end loop

6 close files

end mergeFiles