

实验报告

一、完成度说明

基本功能全部实现

二、详细说明

(一)、用邻接表来完成从任意机场出发的遍历，包括深度优先遍历和广度优先遍历

1.数据结构:

使用邻接表存储，且直接使用机场 ID 作为下标访问，所以时间复杂度为 $O(n+e)$ 。因为是有向图，所以空间复杂度为 $O(n+e)$

2.算法:

基本思想：深度优先遍历用递归实现，由于是从某一特定顶点出发，所以只要求该顶点所在的连通分量即可。广度优先遍历用队列实现，利用队列先进先出的特点。

时间复杂度：由于用邻接表作为存储结构，深度优先遍历查找邻接点所需时间为 $O(e)$ ，所以时间复杂度为 $O(n+e)$ 。广度优先同理，也是 $O(n+e)$ 。

(二)、使用邻接矩阵表来完成任意两个机场的可连通性，包括是否可以直飞、1 次中转、2 次中转等

1.数据结构:

使用邻接矩阵存储，时间复杂度为 $O(e)$ ，空间复杂度为 $O(n^2)$

2.算法:

基本思想：（这里认为中转次数为 k 是限制必须中转 k 次）根据离散数学的知识，邻接矩阵的 k 次幂（其中加法使用逻辑加）得到的矩阵就可以标识两个机场是否能在中转 k 次时联通，所以直接求邻接矩阵的幂并输出即可。而对于没有中转次数限制的情况，调用 1 中的深度优先遍历得到所有可以到达的点，将其在邻接矩阵中的对应位置置为 1 即可。

时间复杂度：对于有限制的情况，调用矩阵乘法，时间复杂度为 $O(n^3)$ ；对于无限制的情况，调用深度优先遍历，但还要遍历邻接矩阵进行输出，时间复杂度为 $O(n^2)$

(三)、求任意两个机场之间的最短飞行时间（含转机停留时间）

1.数据结构:

用 map 类型的数组实现了邻接表,还用到了 STL 中的 set 容器，以及优先队列。map 对应的结构是红黑树，其查找、插入、删除等操作的时间复杂度为 $O(\log n)$ ，二叉堆时间复杂度为 $O(n)$ 。使用 map 构建邻接表，要执行 $n+e$ 次查找、插入等操作，所以空间复杂度为 $O(n+e)$ ，时间复杂度为 $O((n+e)*\log n)$

2.算法:

基本思想：本题需要考虑时序关系，所以拆点建图，新的节点类型包括机场 ID 和所处时刻。用 map 类型数组和 nextedge 类建立邻接表存储，可以实现用结点直接访问对应的边而不必查找。Dijkstra 算法采用优先队列来优化，以其与源点（出发点）的距离为排序依据建立优先队列，那么队首元素就是权值最小的元素，省去了查找的时间。由于出发点有多个出发时刻，所以 for 循环进行遍历，找到所有出发时刻中到达目的顶点的最短用时。

时间复杂度：每次从堆中取出的最小值就是要用的顶点，这样堆中元素共 $O(n)$ 个，更新

和取出的操作有 $O(e)$ 次，因此时间复杂度为 $O(e \cdot \log n)$

(四)、仅限直飞或 1 次中转，求任意两个机场的航线

1. 数据结构：

直飞的情况一边读取一边输出，不涉及数据结构，空间复杂度为 $O(1)$ ，时间复杂度为 $O(n)$ 。一次中转的情况使用优先队列存储，空间复杂度为 $O(n)$ ，时间复杂度为 $O(n \cdot \log n)$

2. 算法：

基本思想：对于直飞的情况，直接在读入表格时判断出发地和目的地是否满足要求，满足要求则直接输出即可。对于一次中转的情况，先用优先队列存下所有航班（按照起飞时间早的优先）然后从队首元素开始依次出队，直至找到从给定出发地出发的航班为止，将其记为 a ，复制该优先队列，在第二层循环中，先将队列元素依次出队直至出发时间晚于 a 的出发时间，然后寻找出发地是 a 的目的地，且目的地为给定目的地的航班 b 。由于该队列是时间优先，此时航班 ab 组成的航班既满足要求，又在时间上合法，是答案之一。重复这个过程得到所有航班。

时间复杂度：由于共有两层循环嵌套，每次从堆中取出的最小值的时间复杂度为 $O(\log n)$ ，所以时间复杂度为 $O((n \cdot \log n)^2)$

(五)、给定起飞时段或者降落时段或者机型要求，求任意两个机场的多个备选航线

1. 数据结构：

存储结构与问题 3 相同，不再赘述，空间复杂度为 $O(n+e)$ ，时间复杂度为 $O((n+e) \cdot \log n)$

2. 算法：

基本思想：本题考虑时序关系，拆点建图，存储结构与问题 3 相同，只是在读取表格数据时就利用所给的不同的限制条件进行读取，只读取符合要求的。（这里认为起飞时段和降落时段的限制是对所有航班的，也包括中转过程中的航班）。在寻找航线是否存在是，由于只需要找到一条航线即可，所以用广度优先搜索（BFS）从指定出发地开始搜索，直到第一次找到目的地即可跳出循环，比较其中转次数是否满足所给条件，满足才输出路径。

时间复杂度：使用队列进行广度优先遍历，时间复杂度为 $O(n+e)$ 。

(六)、给定起飞时段或者降落时段或者机型要求，求任意两个机场之间的航费（机票价格）最低的路径

1. 数据结构：

存储结构与问题 3 相同，不再赘述，空间复杂度为 $O(n+e)$ ，时间复杂度为 $O((n+e) \cdot \log n)$

2. 算法：

基本思想：图的读取、存储都与问题 5 相同，共用一个函数进行读取。搜索航费最低的路径，实质还是用 dijkstra 算法，只是权值变成了航费，同一机场的不同时刻对应的不同点之间权值为 0。dijkstra 算法可以套用问题 3 中的，但为了输出路径，另添 map 结构的 path 数组，来存放路径以便输出。

时间复杂度：使用的算法与问题 3 相同，所以不再重复分析过程，时间复杂度为 $O(e \cdot \log n)$

三、样例输出

```
input:
1 : DFS or BFS traverses
2 : judge if two airports are connected
3 : the minTime between two airports
4 : the flight(direct or transit 1) between two airports
5 : the flight(transit less than k times) with limits
6 : the flight(minFare) with limits
1
邻接表进行深度优先和广度优先遍历，请输入出发机场的ID
1
深度优先遍历：
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
广度优先遍历：
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
=====
Process exited after 8.32 seconds with return value 0
请按任意键继续. . .
```

```

input:
1: DFS or BFS traverses
2: judge if two airports are connected
3: the minTime between two airports
4: the flight(direct or transit 1) between two airports
5: the flight(transit less than k times) with limits
6: the flight(minFare) with limits
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:

```

```
input:
1 : DFS or BFS traverses
2 : judge if two airports are connected
3 : the minTime between two airports
4 : the flight(direct or transit 1) between two airports
5 : the flight(transit less than k times) with limits
6 : the flight(minFare) with limits
3
请输入出发地和目的地
48 50
165

-----
Process exited after 9.871 seconds with return value 0
请按任意键继续. . .
```

4.

```
D:\编程\未命名1.exe
input:
1 : DFS or BFS traverses
2 : judge if two airports are connected
3 : the minTime between two airports
4 : the flight(direct or transit 1) between two airports
5 : the flight(transit less than k times) with limits
6 : the flight(minFare) with limits
4
请输入出发地和目的地，以及中转次数
48 49 1
1697 916
1697 1049
1697 2319
1697 2321
1826 2198
1826 2199
1826 2200
1843 2200
5
-----
Process exited after 13.86 seconds with return value 0
请按任意键继续...
```

5.

```
D:\编程\未命名1.exe
input:
1 : DFS or BFS traverses
2 : judge if two airports are connected
3 : the minTime between two airports
4 : the flight(direct or transit 1) between two airports
5 : the flight(transit less than k times) with limits
6 : the flight(minFare) with limits
5
input:
1 : limit departure time
2 : limit arrival time
3 : limit airplane model
3
请输入出发机场，目的机场和中转次数上限，以及机型要求
48 50 1 1
85
-----
Process exited after 15.92 seconds with return value 0
请按任意键继续...
```

6.

```
input:
1 : DFS or BFS traverses
2 : judge if two airports are connected
3 : the minTime between two airports
4 : the flight(direct or transit 1) between two airports
5 : the flight(transit less than k times) with limits
6 : the flight(minFare) with limits
6
input:
1 : limit departure time
2 : limit arrival time
3 : limit airplane model
3
请输入出发机场，目的机场和机型要求
48 50 1
1
666
-----
Process exited after 8.516 seconds with return value 0
```