

Cách Học

Cách học lập trình ...

Blog Chuyên mục Android Pascal C/C++

Java Swing Java MySQL Bài tập tư duy

Thông báo blog nguyenvanquan7826.com chuyển tên miền sang cachhoc.net

[Thuật toán] Tìm đường đi ngắn nhất Dijkstra, Floyd



Tweet

Like 0

Share

G+

Update 25/05/2014: Do một số góp ý của các bạn nên mình đã viết thêm 1 chương trình của thuật toán Dijkstra theo cấu trúc hàm và cũng nhân tiện chỉnh lại chút code cho sáng sủa và chính xác hơn ^^.

Update 27/09/2014: bổ xung code pascal của thuật toán tại đây: <http://ideone.com/c7J0dq>

Update

14/06/2014:

Chương trình

mô phỏng

thuật toán

Dijkstra

Nội dung

Thuật toán Dijkstra

Thuật toán Floyd

Code nâng cao cho cả 2 thuật toán

Translate blog

Tiếng Việt

Khóa Học Laravel 5 Từ

QC Lập Trình Web Thực Tế |
Làm Chủ Hoàn Toàn Laravel

Vietpro Academy

Tìm hiểu thêm

Ứng dụng Android, iOS

Liên hệ:

nguyenvanquan7826@gmail.com

Lưu ý: Không nhận các loại bài tập, thực tập của học sinh, sinh viên.

Mã giảm giá
40% Khóa
học online +
unica

Trong bài viết này chỉ đề cập tới các thuật toán tìm đường đi ngắn nhất Dijkstra và Floyd, một số thuật ngữ liên quan mình sẽ không giải thích hay định nghĩa, các bạn tự tìm hiểu trong sách hoặc trên mạng.

Bài toán đường đi ngắn nhất nguồn đơn là bài toán tìm một đường đi giữa hai đỉnh sao cho tổng các trọng số của các cạnh tạo nên đường đi đó là nhỏ nhất. Hay nói một cách toán học là:

Cho đơn đồ thị liên thông, có trọng số $G=(V,E)$. Tìm khoảng cách $d(a,b)$ từ một đỉnh a cho trước đến một đỉnh b bất kỳ của G và tìm đường đi ngắn nhất từ a đến b .

Như tiêu đề bài viết, chúng ta sẽ tìm hiểu 2 thuật toán để giải quyết bằng cách sử dụng mà trận kề của đồ thị (chú ý ta xét trọng số của đồ thị là không âm).

Ma trận kề của đồ thị có n đỉnh là ma trận vuông G có số hàng số cột là n . $G[i][j]$ là độ dài đường đi từ đỉnh i tới đỉnh j . Nếu xét đồ thị vô hướng thì $G[i][j] = G[j][i]$. Độ dài từ một đỉnh tới chính nó luôn là 0 ($G[i][i] = 0$). Nếu giữa 2 cạnh i và j của đồ thị không tồn tại đường đi thì $G[i][j] = \infty$. Tuy nhiên khi biểu diễn trong máy tính thì giá trị ∞ được đặt là 1 hằng số rất lớn hoặc là tổng các giá trị trong ma trận (tổng độ dài các cạnh).

1. Thuật toán Dijkstra

Về thuật toán Dijkstra có 2 loại là tìm đường đi ngắn nhất từ 1 đỉnh nguồn tới 1 đỉnh đích và tìm đường đi ngắn nhất từ 1 đỉnh nguồn tới các đỉnh còn lại của đồ thị, và ở đây mình sẽ nói về loại thứ 1. (Loại thứ hai bạn có thể tìm trên mạng hoặc chỉ cần thay đổi dòng **while** (**s[b] == 0**) (dòng 43 của code 1 & dòng 76 của code 2) thành vòng **for** duyệt từ 0 đến $n-1$ là sẽ tìm được tất cả các đỉnh).

- Dùng 1 mảng $Len[]$ - $Len[i]$ là khoảng cách ngắn nhất từ đỉnh a tới đỉnh i .

- Dùng 1 mảng S đánh dấu các đỉnh i đặc biệt (các đỉnh i mà thời điểm hiện tại thì đường đi từ a tới i là ngắn nhất).



Bài được xem nhiều

- [School - Pháp luật] Bài tập chia thừa kế
- [School_PPNCKH] Hệ thống các phương pháp nghiên cứu khoa học
- Lập trình C: Bài 3 - Phép toán, toán tử trong C
- [Cây] Một số phép toán trên cây nhị phân tìm kiếm
- Lập trình C: Bài 2 - Kiểu dữ liệu và nhập xuất trong C
- [Thuật toán] Tìm đường đi ngắn nhất Dijkstra, Floyd
- Lập trình C: Bài 13 - Danh sách liên kết đơn cài bằng con trỏ
- [Pascal - TUT] Bài 7: Chương trình con
- Lập trình C: Bài 11 - Kiểu cấu trúc - struct
- [Pascal - TUT] Bài 9: Kiểu bản ghi - Record

- Dùng mảng $P[]$ đánh dấu đường đi. $P[j] = i$ nếu i là đỉnh đi trước j trong đường đi ngắn nhất.
- Đặt lại giá trị vô cùng cho các cặp đỉnh không có đường đi.
- Khởi tạo tất cả các đường đi từ a đến các đỉnh khác bằng vô cùng.
- Khởi tạo đường đi từ a đến chính $a = 0$.
- Duyệt hết các đỉnh V của đồ thị
 - + Tìm đỉnh i chưa nằm trong S mà đường đi từ a tới i là ngắn nhất để đưa vào S . Nếu không tìm được đỉnh nào nghĩa là đã duyệt hết các đỉnh có thể đi mà vẫn chưa thấy đỉnh đích \Rightarrow không thể đi được.
 - + Nếu tìm được đỉnh i thì duyệt tất cả các đỉnh j chưa nằm trong S . Nếu $Len[i] + G[i][j] < Len[j]$ (trong đó $G[i][j]$ là khoảng cách từ đỉnh i tới đỉnh j) thì gán $Len[j] = Len[i] + G[i][j]$; và đánh dấu đường đi $P[j] = i$.

Lưu ý: Do trong C , mảng bắt đầu từ 0. Do vậy các đỉnh khi tính toán thì sẽ tính từ đỉnh 0 đến đỉnh $n-1$. Tuy nhiên khi hiển thị ra thì vẫn phải là từ đỉnh 1 đến n và trong file `input.inp` thì đỉnh đầu và đỉnh cuối cũng sẽ được tính từ 1 đến n . Do đó trong code trước khi tính toán ta cần giảm đỉnh đầu và đỉnh cuối đi 1 đơn vị. Sau khi tính toán xong thì khi xuất kết quả lại cần tăng các đỉnh trong đường đi tìm được lên 1 đơn vị để hiển thị đúng (VD ta muốn tính đường đi từ đỉnh 4 đến đỉnh 8, thì đỉnh 4 tương ứng với vị trí thứ 3 trong mảng, đỉnh 8 ứng với vị trí thứ 7 nên ta cần giảm 4 xuống 3, 8 xuống 7 để tính toán. Khi tìm được đường đi, giả sử là $3 \rightarrow 5 \rightarrow 4 \rightarrow 7$ thì phải in ra là $4 \rightarrow 6 \rightarrow 5 \rightarrow 8$).

Chúng ta sẽ đi thực hành với đồ thị sau theo 2 code là làm ngay trong main và thực hiện theo các hàm:

Theo dõi blog qua email

Join 168 other subscribers

Địa chỉ thư điện tử Theo dõi

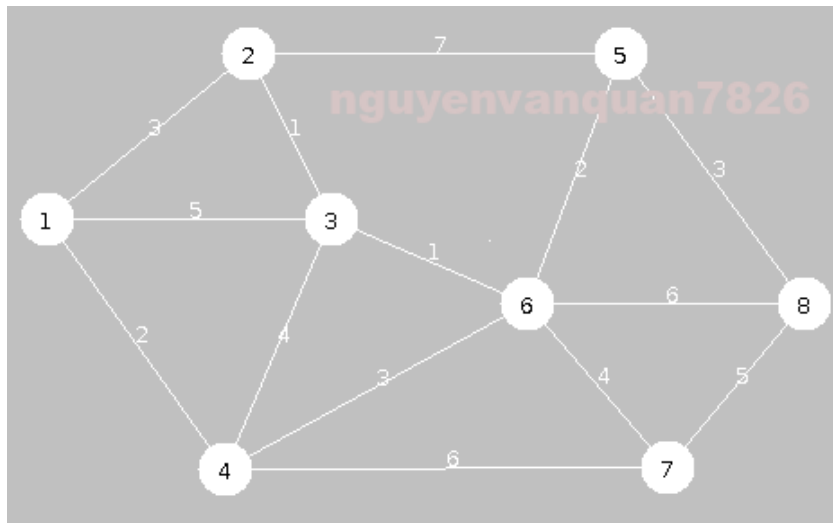


CẢNH BÁO: SAU KHI HỌC NÀY BẠN SẼ CUỒNG EXCEL

Chỉ cần 30 phút mỗi ngày | nhà bạn sẽ giỏi Excel sau

edu mail

Tìm Hi



file **input.inp**: hàng đầu tiên thể hiện có 8 điểm, đi từ điểm 4 đến điểm 8. ma trận 8×8 ở dưới là ma trận kề của đồ thị.

```

1 | 8 4 8
2 | 0 3 5 2 0 0 0 0
3 | 3 0 1 0 7 0 0 0
4 | 5 1 0 4 0 1 0 0
5 | 2 0 4 0 0 3 6 0
6 | 0 7 0 0 0 2 0 3
7 | 0 0 1 3 2 0 4 6
8 | 0 0 0 6 0 4 0 5
9 | 0 0 0 0 3 6 5 0

```

* Code ngay trong main

Code này đã được sửa và khắc phục một số lỗi từ [code ngày trước](#) (hoặc [link dự phòng](#)).

```

01 | #include <stdio.h>
02 | #include <stdlib.h>
03 | #define INP "input.inp"
04 | #define OUT "output.out"
05 |
06 | int main() {
07 |     FILE *fi = fopen(INP, "r");
08 |     FILE *fo = fopen(OUT, "w");
09 |     int n, a, b, i, sum = 0;
10 |
11 |     // nhập dữ liệu từ file input
12 |     fscanf(fi, "%d%d%d", &n, &a, &b);
13 |     int G[n][n];
14 |     int S[n], Len[n], P[n];
15 |
16 |     // nhập ma trận và tính giá trị vào cu
17 |     for (i = 0; i < n; i++)
18 |         for (int j = 0; j < n; j++) {
19 |             fscanf(fi, "%d", &G[i][j]);
20 |             sum += G[i][j];
21 |         }

```

```

22 // dat vo cung cho tat ca cap canh kt
23 for (int i = 0; i < n; i++) {
24     for (int j = 0; j < n; j++) {
25         if (i != j && G[i][j] == 0)
26             G[i][j] = sum;
27     }
28 }
29
30 /* Do mang tinh tu G[0][0] nen can gi
31    di 1 don vi de tinh toan cho phu hop
32 a--;
33 b--;
34
35 for (int i = 0; i < n; i++) {
36     Len[i] = sum;
37     S[i] = 0;
38     P[i] = a;
39 }
40
41 Len[a] = 0;
42
43 // tim duong di ngan nhat tu 1 dinh c
44 //for (int k = 0; k < n; k++)
45 while (S[b] == 0) {
46     for (i = 0; i < n; i++)
47         if (!S[i] && Len[i] < sum)
48             break;
49
50     // i >= n tuc la duyet het cac dir
51     if (i >= n) {
52         printf("done dijkstra\n");
53         break;
54     }
55
56     for (int j = 0; j < n; j++) {
57         if (!S[j] && Len[i] > Len[j])
58             i = j;
59     }
60 }
61
62 S[i] = 1;
63
64 for (int j = 0; j < n; j++) {
65     if (!S[j] && Len[i] + G[i][j] < Len[j])
66         Len[j] = Len[i] + G[i][j];
67     P[j] = i;
68 }
69 }
70 }
71
72 printf("done dijkstra\n");
73
74 /* Do ta dang tinh toan tu dinh 0 ner
75    muon hien thi tu dinh 1 thi can dung
76
77 printf("start find path\n");
78

```

```

79     if (Len[b] > 0 && Len[b] < sum) {
80         fprintf(fo, "Length of %d to %d i
81
82         // truy vet
83         while (i != a) {
84             fprintf(fo, "%d <-- ", i + 1);
85             i = P[i];
86         }
87         fprintf(fo, "%d", a + 1);
88     } else {
89         fprintf(fo, "khong co duong di tu
90     }
91
92     printf("done find path\n");
93
94     fclose(fi);
95     fclose(fo);
96
97     printf("done - open file output to se
98     return 0;
99 }

```

[link dự phòng](#)

* Code theo từng hàm

Trong code theo hàm có hàm:

- **readData** thực hiện đọc thông tin từ file input.
- **dijkstra** thực hiện thuật toán
- **back** thực hiện trả về chuỗi là đường đi tìm được
- **outResult** thực hiện in ra file output kết quả

```

001 #include <stdio.h>
002 #include <stdlib.h>
003 #include <cstring>
004
005 #define INP "input.inp"
006 #define OUT "output.out"
007
008 // read data in file input
009 int readData(int ***G, int *n, int *a, i
010     FILE *fi = fopen(INP, "r");
011     if (fi == NULL) {
012         printf("file input not found!\n'
013         return 0;
014     }
015     printf("start read file\n");
016
017     fscanf(fi, "%d %d %d", n, a, b);
018
019     *G = (int **) malloc((*n) * sizeof(i
020     for (int i = 0; i < *n; i++) {
021         (*G)[i] = (int *) malloc((*n) *
022         for (int j = 0; j < *n; j++) {
023             int x;

```

```

024         fscanf(fi, "%d", &x);
025         (*G)[i][j] = x;
026     }
027 }
028
029 fclose(fi);
030 printf("done read file\n");
031 return 1;
032 }
033
034 // thuật toán dijkstra
035 int dijkstra(int **G, int n, int a, int
036
037     /* Do mảng tính từ G[0][0] nên cần g
038     đi 1 đơn vị để tính toán cho phù h
039     a--;
040     b--;
041
042     printf("start dijkstra\n");
043
044     int* Len = (int *) malloc(n * sizeof
045     int* S = (int *) malloc(n * sizeof(i
046
047     int sum = 0;           // giá trị \
048
049     // tính giá trị vô cùng (sum)
050     for (int i = 0; i < n; i++) {
051         for (int j = 0; j < n; j++) {
052             sum += G[i][j];
053         }
054     }
055
056     // đặt vô cùng cho tất cả các cạnh k
057     for (int i = 0; i < n; i++) {
058         for (int j = 0; j < n; j++) {
059             if (i != j && G[i][j] == 0)
060                 G[i][j] = sum;
061         }
062     }
063
064     for (int i = 0; i < n; i++) {
065         Len[i] = sum;           // khởi tạo
066         S[i] = 0;              // danh sách
067         P[i] = a;              // đặt điểm
068     }
069
070     Len[a] = 0;               // đặt độ dài
071
072     int i;
073
074     // tìm đường đi ngắn nhất từ 1 đỉnh
075     //for (int k = 0; k < n; k++)
076     while (S[b] == 0) {
077         for (i = 0; i < n; i++)
078             if (!S[i] && Len[i] < sum)
079                 break;
080

```

```

081     // i >= n tức là duyệt hết các đỉnh
082     if (i >= n) {
083         printf("done dijkstra\n");
084         return 0;
085     }
086
087     for (int j = 0; j < n; j++) {
088         if (!S[j] && Len[i] > Len[j])
089             i = j;
090     }
091
092     S[i] = 1;
093
094     for (int j = 0; j < n; j++) {
095         if (!S[j] && Len[i] + G[i][j] < Len[j])
096             Len[j] = Len[i] + G[i][j];
097         P[j] = i;
098     }
099 }
100 }
101 printf("done dijkstra\n");
102 return Len[b];
103 }
104
105 // truy vết đường đi
106 void back(int a, int b, int *P, int n, char *path) {
107
108     //char *path = (char *) malloc((n * sizeof(char) + 1));
109
110     /* Do mảng tính từ G[0][0] nên cần giảm
111        đi 1 đơn vị để tính toán cho phù hợp */
112     a--;
113     b--;
114
115     printf("start find path\n");
116
117     int i = b;
118     int point[n]; // danh sách các đỉnh
119     int count = 0;
120
121     /* Do ta đang tính toán từ đỉnh 0 nên
122        muốn hiển thị từ đỉnh 1 thì cần dư
123        1 đơn vị */
124     point[count++] = i + 1;
125     while (i != a) {
126         i = P[i];
127         point[count++] = i + 1;
128     }
129
130     strcpy(path, "");
131     char temp[10];
132     for (i = count - 1; i >= 0; i--) {
133         sprintf(temp, "%d", point[i]);
134         strcat(path, temp);
135
136         if (i > 0) {
137             sprintf(temp, " --> ");

```



```

138         strcat(path, temp);
139     }
140 }
141
142 printf("done find path\n");
143 }
144
145 void outResult(int len, char* path) {
146     FILE *fo = fopen(OUT, "w");
147
148     if (len > 0) {
149         fprintf(fo, "\nLength of %c to %",
150             path[strlen(path) - 1],
151         }
152
153         fprintf(fo, "path: %s\n", path);
154
155         fclose(fo);
156     }
157
158     int main() {
159         int **G, n, a, b, len;
160
161         if (readData(&G, &n, &a, &b) == 0) {
162             return 0;
163         }
164
165         char *path = (char *) malloc((10 * n
166         int P[n];
167
168         len = dijkstra(G, n, a, b, P);
169
170         if (len > 0) {
171             back(a, b, P, n, path);
172             outResult(len, path);
173         } else {
174             char *path = (char *) malloc((n
175             sprintf(path, "khong co duong di
176             outResult(len, path);
177         }
178
179         printf("done - open file output to s
180         return 0;
181     }

```

[link dự phòng](#)

Nhìn code có vẻ hơi dài nhưng khi đọc hiểu rồi thì chả dài tẹo nào @@ =)).

2. Thuật toán Floyd

Thuật toán này cho phép chúng ta tìm đường đi ngắn nhất giữa mọi cặp đỉnh.



Nếu đỉnh k nằm trên đường đi ngắn nhất từ đỉnh i tới đỉnh j thì đoạn đường từ i tới k và từ k tới j phải là đường đi ngắn nhất từ i tới k và từ k tới j tương ứng. Do đó ta sử dụng ma trận A để lưu độ dài đường đi ngắn nhất giữa mọi cặp đỉnh.

– Ban đầu ta đặt $A[i,j] = C[i,j]$, tức là ban đầu A chứa độ dài đường đi trực tiếp (không đi qua đỉnh nào cả).

– Sau đó thực hiện n lần lặp, sau lần lặp thứ k , ma trận A sẽ chứa độ dài đường đi ngắn nhất giữa mọi cặp đỉnh chỉ đi qua các đỉnh thuộc tập $\{1,2,...,k\}$. Như vậy, sau n lần lặp ta nhận được ma trận A chứa độ dài các đường đi ngắn nhất giữa mọi cặp đỉnh của đồ thị.

– Ký hiệu A_k là ma trận A sau lần lặp thứ k , tức là $A_k[i,j]$ là độ dài đường đi ngắn nhất từ i đến j chỉ đi qua các đỉnh thuộc $\{1, 2,..., k\}$. $A_k[i,j]$ được tính theo công thức như sau: $A_k[i,j] = \min \{A_{k-1}[i,j], A_{k-1}[i,k] + A_{k-1}[k,j]\}$.

– Trong quá trình lặp ta phải lưu lại vết đường đi, tức là đường đi ngắn nhất đi qua các đỉnh nào. Khi đó ta sử dụng mảng phụ $P[n \times n]$, trong đó $P[i,j]$ lưu đỉnh k nếu đường đi ngắn nhất từ i đến j đi qua đỉnh k . Ban đầu $P[i,j]=0$ với mọi i,j , vì lúc đó đường đi ngắn nhất là đường đi trực tiếp, không đi qua đỉnh nào cả.

Code thuật toán:

```

01 void Floyd (int a, int b)
02 {
03     int max = tongthiethai();
04     for (int i=0; i<n; i++)
05         for (int j=0; j<n; j++)
06             {
07                 if (G[i][j])
08                     A[i][j] = G[i][j];
09                 else A[i][j] = max;
10                 P[i][j] = -1;
11             }
12
13     for (int k=0; k<n; k++) // lap n la
14     {
15         for (int i=0; i<n; i++) // thay
16             for (int j=0; j<n; j++)
17                 if (A[i][j] > A[i][k] + A[k][j])
18                     {
19                         A[i][j] = A[i][k] + A[k][j];
20                         P[i][j] = k ;
21                     }
22     }

```

23 | }

Cách xây dựng chương trình hoàn chỉnh hoàn toàn giống với thuật toán Dijkstra.

Code nâng cao

Đây là code cho phép chọn 1 trong 2 thuật toán trên và xuất ra file đúng theo quá trình làm như các kết quả trong hình bên dưới hoặc link dự phòng

file input.inp:

```
8
A B C D E F G H
0 3 5 2 0 0 0 0
3 0 1 0 7 0 0 0
5 1 0 4 0 1 0 0
2 0 4 0 0 3 6 0
0 7 0 0 0 2 0 3
0 0 1 3 2 0 4 6
0 0 0 6 0 4 0 5
0 0 0 0 3 6 5 0
```



Menu console

```
Ma tran ke cua do thi
A B C D E F G H
0 3 5 2 0 0 0 0
3 0 1 0 7 0 0 0
5 1 0 4 0 1 0 0
2 0 4 0 0 3 6 0
0 7 0 0 0 2 0 3
0 0 1 3 2 0 4 6
0 0 0 6 0 4 0 5
0 0 0 0 3 6 5 0

nguyenvanquan7826

Moi ban chon thuat toan :
1: Thuat toan Dijkstra
2: Thuat toan Floyd
3: Thoat !
1

----Thuat toan Dijkstra----

Cac dinh danh so tu 1 den 8.Hoac tu A den H
Nhap dinh bat dau : 2
Nhap dinh ket thuc : 6
Hoan thanh ! Mo file output.out de xem ket qua !Press any key to continue . . .

Moi ban chon thuat toan :
1: Thuat toan Dijkstra
2: Thuat toan Floyd
3: Thoat !
2

----Thuat toan Floyd----

Cac dinh danh so tu 1 den 8.Hoac tu A den H
Nhap dinh bat dau : 3
Nhap dinh ket thuc : 5
Hoan thanh ! Mo file output.out de xem ket qua !Press any key to continue . . .

Moi ban chon thuat toan :
1: Thuat toan Dijkstra
2: Thuat toan Floyd
3: Thoat !
-
```

Output Dijkstra

Ma tran ke cua do thi

A B C D E F G H

```
0 3 5 2 0 0 0 0
3 0 1 0 7 0 0 0
5 1 0 4 0 1 0 0
2 0 4 0 0 3 6 0
0 7 0 0 0 2 0 3
0 0 1 3 2 0 4 6
0 0 0 6 0 4 0 5
0 0 0 0 3 6 5 0
```

Thuật toán Dijkstra

TT	1 (A)	2 (B)	3 (C)	4 (D)	5 (E)	6 (F)	7 (G)	8 (H)
1	[~,~]	*[0,2]	[~,~]	[~,~]	[~,~]	[~,~]	[~,~]	[~,~]
2	[3,2]	-	*[1,2]	[~,2]	[7,2]	[~,2]	[~,2]	[~,2]
3	[3,2]	-	-	[5,3]	[7,2]	*[2,3]	[~,2]	[~,2]
4	*[3,2]	-	-	[5,3]	[4,6]	-	[6,6]	[8,6]
5	-	-	-	[5,3]	*[4,6]	-	[6,6]	[8,6]
6	-	-	-	*[5,3]	-	-	[6,6]	[7,5]
7	-	-	-	-	-	-	*[6,6]	[7,5]
8	-	-	-	-	-	-	-	*[7,5]

Do dai ngan nhat cua duong di tu 2(B) den 6(F) la 2
 Qua trinh duong di: 2 --> 3 --> 6

Output Floyd

Bước thứ 6

Thuật toán Floyd

Ma trận Floyd								P							
4	3	4	2	7	5	8	10	4	0	2	0	6	3	4	6
3	2	1	5	4	2	6	7	0	3	0	1	6	3	6	6
4	1	2	4	3	1	5	6	2	0	2	0	6	0	6	6
2	5	4	4	5	3	6	8	0	1	0	1	6	0	0	6
7	4	3	5	4	2	6	3	6	6	6	6	6	0	6	0
5	2	1	3	2	2	4	5	3	3	0	0	0	3	0	5
8	6	5	6	6	4	8	5	4	6	6	0	6	0	6	0
10	7	6	8	3	5	5	6	6	6	6	6	0	5	0	5

Bước thứ 7

nguyenvanquan7826

			A								P							
4	3	4	2	7	5	8	10			4	0	2	0	6	3	4	6	
3	2	1	5	4	2	6	7			0	3	0	1	6	3	6	6	
4	1	2	4	3	1	5	6			2	0	2	0	6	0	6	6	
2	5	4	4	5	3	6	8			0	1	0	1	6	0	0	6	
7	4	3	5	4	2	6	3			6	6	6	6	6	0	6	0	
5	2	1	3	2	2	4	5			3	3	0	0	0	3	0	5	
8	6	5	6	6	4	8	5			4	6	6	0	6	0	6	0	
10	7	6	8	3	5	5	6			6	6	6	6	0	5	0	5	

Đo dài ngắn nhất của đường đi từ 7(G) đến 5(E) là 6

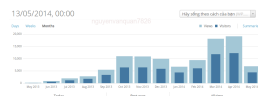
Qua trình đường đi: 7 --> 6 --> 5

Có liên quan



[Java - Thuật toán]
Mô phỏng thuật toán
Dijkstra tìm đường đi
ngắn nhất

Tháng Sáu 14, 2014
Trong "TT Đồ thị"



[Tự sướng] Kỷ niệm
Blog tròn 1 tuổi
Tháng Năm 13, 2014
Trong "Cuộc sống
muôn màu"

Thuật toán game
pokemon (pikachu)
Nội dung Trường
hợp cùng nằm trên
một hàng hoặc cột
Trường hợp đi theo
chiều ngang, dọc
Tháng Ba 25, 2014
Trong "Thuật toán"

📅 Tháng Mười 13, 2013 👤 nguyenvanquan7826 📁 TT

Đồ thị 💬 113 responses 📖 Dijkstra, đường đi ngắn nhất,
Floyd, thuật toán Dijkstra, thuật toán Floyd, thuật toán tìm
đường đi ngắn nhất

Additional Reading...

- [Java - Thuật toán] Mô phỏng thuật toán Dijkstra tìm đường đi ngắn nhất

- [Thuật toán – Java] Chương trình mô phỏng thuật toán tìm kiếm theo chiều rộng (BFS) và chiều sâu (DFS)

113 responses on [Thuật toán] Tìm đường đi ngắn nhất Dijkstra, Floyd

Pingback: [\[Tự sướng\] Kỷ niệm Blog tròn 1 tuổi | Chia sẻ để cuộc sống tốt đẹp hơn!](#)



Thái viết:

Tháng Sáu 13, 2014 lúc 5:45 chiều

cho em hỏi sao em chạy mọi đồ thị đều không tìm được đường đi vậy. nhưng giai tay thì vẫn có.

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Sáu 13, 2014 lúc 10:35 chiều

@@ Bạn xem kỹ lại code với file input nhé. Mình chạy ok.

[Trả lời](#)



Rias Gremory viết:

Tháng Năm 11, 2015 lúc 8:30 sáng

Bạn ơi sao copy code full cuối bài về khi chạy bị lỗi này

[Error]

C:\Users\Admin\Downloads\ideone_DpaGPW.cpp:167:

E2313 Constant expression required in function

Dijkstra(GRAPH,int,int)

[Error]

C:\Users\Admin\Downloads\ideone_DpaGPW.cpp:283:

E2313 Constant expression required in function

floyd(GRAPH,int,int)

Trả lời



nguyenvanquan7826 viết:

Tháng Năm 11, 2015 lúc 9:21 sáng

Bạn copy code nào vậy? gửi lại file cho mình vào mail nguyenvanquan7826@gmail.com mình xem cho nhé.

Trả lời



Rias Gremory viết:

Tháng Năm 12, 2015 lúc 2:35 chiều

mình gửi mail rồi đó



nguyenvanquan7826 viết:

Tháng Năm 12, 2015 lúc 3:55 chiều

Mình nhận được rồi, mình lấy code đó chạy vẫn ổn, không vấn đề gì. Có lẽ bạn dùng C-free nên nó không được chuẩn lắm hoặc lỗi gì đó. Bạn bật teamview mình xem cho nhé. Liên lạc với mình qua skype: nguyenvanquan7826



Hữu Tín viết:

Tháng Mười Hai 13, 2017 lúc 10:18 chiều

a sao ko có đường đi ngắn nhất a @@ ví dụ 1->4=1<-2<-3<-4

Trả lời



nguyenvanquan7826 viết:

Tháng Mười Hai 13, 2017 lúc 10:23 chiều

Ah vì trong bài mình ví dụ đi từ 1->8 mà. đâu có đi đến 4 đâu.

[Trả lời](#)



Le Thi Bich Ngoc viết:

Tháng Mười Một 14, 2017 lúc 4:13 chiều

SAO em chạy chương trình nó hiện ra nó found input file mà em đã để file .cpp va file .inp vào chung 1 folder rồi. Cảm ơn anh

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Mười Một 14, 2017 lúc 6:21 chiều

Bạn xem lại đuôi file nhé, có thể đuôi file của bạn không phải là .inp mà là .txt bị ẩn.

Bạn cho hiện đuôi file lên để kiểm tra ở đây với win 7, ở đây với win 10 nhé.

[Trả lời](#)

Pingback: [\[Java - Thuật toán\] Mô phỏng thuật toán Dijkstra | Chia sẻ để cuộc sống tốt đẹp hơn!](#)

Pingback: [\[Java - Thuật toán\] Mô phỏng thuật toán Dijkstra tìm đường đi ngắn nhất | Chia sẻ để cuộc sống tốt đẹp hơn!](#)



itmietvuon viết:

Tháng Chín 23, 2014 lúc 2:51 chiều

Bạn có thể đổi code sang Pascal được không vậy? Biết 99% câu trả lời sẽ là "Mình đang bận lắm!" nhưng vẫn hy vọng bạn sẽ chuyển thành 1 bản code bằng Pascal để phổ biến kiến thức nhiều hơn đến các newbie không

rành về C.
Cám ơn bạn đã đọc.

Trả lời



nguyenvanquan7826 viết:

Tháng Chín 23, 2014 lúc 11:41 chiều

Mình sẽ cố gắng chuyển đổi code sang pascal trong tuần này nhé. Cảm ơn bạn đã quan tâm tới các bài viết trong blog và đặc biệt là tinh thần chia sẻ của bạn 😊

Trả lời



nguyenvanquan7826 viết:

Tháng Chín 27, 2014 lúc 2:35 sáng

*Bạn có thể xem code pascal tại đây nhé
<http://ideone.com/c7J0dq>*

Trả lời



itmietvuon viết:

Tháng Chín 28, 2014 lúc 4:36 sáng

Cám ơn bạn rất nhiều, mong VN sẽ ngày mỗi nhiều hơn những người như bạn.

Chúc bạn sớm có thêm nhiều bài viết hay cho cộng đồng những người yêu lập trình.

Trả lời



nguyenvanquan7826 viết:

Tháng Chín 28, 2014 lúc 11:37 chiều

Cám ơn bạn 😊 Mình sẽ cố gắng viết tốt hơn 😊

Trả lời

**Chile** viết:

Tháng Mười Một 25, 2014 lúc 7:57 sáng

Anh ơi giúp em bài này với! Em đang làm tiểu luận môn học, đề bài của em như sau:

Đồ thị vô hướng $G=\{V,E\}$ được cho bởi danh sách cạnh DS. Cho u, v thuộc V . Xây dựng thuật toán tìm hai đường đi $A1(u,v)$ và $A2(u,v)$ sao cho không có cạnh nào chung và có tổng độ dài ngắn nhất.

Em loay hoay mãi mà không biết bắt đầu từ đâu nữa ạ. Mong anh giúp dùm em.

[Trả lời](#)**nguyenvanquan7826** viết:

Tháng Mười Một 25, 2014 lúc 3:25 chiều

Cái này bản chất là tìm đường đi ngắn nhất mà.

$A1 + A2$ ngắn nhất khi $A1$ ngắn nhất và $A2$ ngắn nhất.

Bạn dùng dijkstra tìm $A1, A2$, trong quá trình tìm $A2$ thì mỗi lần tìm được 1 cạnh phải xem cạnh đó có trong $A1$ chưa nhé.

[Trả lời](#)**Chile** viết:

Tháng Mười Một 26, 2014 lúc 4:27 chiều

thanks anh ạ

[Trả lời](#)**bd** viết:

Tháng Mười Một 26, 2014 lúc 12:00 chiều

thuật toán này anh đã làm vs Heap chưa ạ? Em ko rõ Dijkstra Heap lắm. Nếu chưa, anh có thể làm hay chỉ cho e đc k ạ?

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Mười Một 26, 2014 lúc 3:15 chiều

Cái này anh chưa làm với Heap. Cũng chưa nghịch Heap bao giờ lun :D. Em thử search trên mạng xem có không 😊

[Trả lời](#)



bd viết:

Tháng Mười Một 26, 2014 lúc 4:34 chiều

vâng ạ. Anh cho em hỏi thêm. Với loại 2, em muốn tìm từ 1 đỉnh tới các đỉnh thì ngoài thay thành vòng for thì em còn phải đổi những j nữa ạ, như phần input em cũng k biết phải ghi sao. Ngoài ra, em ko chạy đc code theo từng phần của anh ạ. Nó chạy tới phần start dijkstra thì crush ạ. Anh có thể chỉ cho e đc k ạ?

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Mười Một 26, 2014 lúc 6:16 chiều

Khi chạy từ 1 đỉnh tới mọi đỉnh khác thì chỉ thay vòng for duyệt hết toàn bộ các đỉnh là xong, không cần thêm gì khác.

Code theo từng hàm thì bạn cứ xây dựng bình thường là được mà.

[Trả lời](#)



bd viết:

Tháng Mười Một 26, 2014 lúc 8:50 chiều

vâng em cảm ơn



huyền viết:

Tháng Mười Một 26, 2014 lúc 9:44 chiều

bạn ơi cho mình hỏi tí..mình có làm đồ án về cái này mà dùng java bạn có thể giải thích rõ hơn về code phần java của bạn đc ko

[Trả lời](#)



huyền viết:

Tháng Mười Một 26, 2014 lúc 9:45 chiều

quên,,mình cần lắm có gì bạn giúp mình với nhá..mình cảm ơn bạn nhiều

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Mười Một 26, 2014 lúc 11:28 chiều

Bạn cứ làm đi, nếu được mình sẽ giúp :v

[Trả lời](#)



huyền viết:

Tháng Mười Một 27, 2014 lúc 11:43 sáng

bạn có thể giải thích dùng mình code phần chạy dãy thuật đc ko :(..bạn cho mình sdt bạn đi

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Mười Một 27, 2014 lúc 12:00 chiều

*Chạy dãy thuật là sao bạn?
có gì bạn có thể gọi vào số của mình.
096.567.7826*

[Trả lời](#)



huyen viết:

Tháng Mười Một 27, 2014 lúc 11:45 sáng

thấy bạn trl cmt mà rơi nước mắt vì mừng 😊

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Mười Một 27, 2014 lúc 12:00 chiều

Làm gì đến mức đó 😊

[Trả lời](#)



doveandrose viết:

Tháng Mười Hai 1, 2014 lúc 5:50 chiều

chào bạn Nguyễn Văn Quân , mình có thể mời bạn cho biết đáp số của bài toán sau đây dc ko ?

cho hệ trục tọa độ Oxy , có các điểm sau đây

A1(586,363),A2(254,137),A3(467,516),A4(798,472),A5(599,213),A6(372,344),A7(146,412),

A8(818,346),A9(850,199),A10(314,260),A11(72,268),A12(731,57),A13(429,179),A14(499,81),

A15(89,169),A16(113,82),A17(904,59),A18(926,551),A19(54,20)

Giả sử rằng 2 điểm bất kỳ đều có đoạn nối và vô hướng
Độ dài đoạn nối (kiểu số thực) giữa 2 điểm bất kỳ dc tính theo kiểu toán phổ thông đã học

Điểm bắt đầu : A18(926,551)

Cho biết độ dài đường đi (kiểu số thực) ngắn nhất xuất phát từ A18 và đi qua tất cả các đỉnh còn lại (mỗi đỉnh phải đi qua đúng 1 lần)

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Mười Hai 1, 2014 lúc 10:44 chiều

Với bài này nó lại là một thuật toán khác rồi bạn. Mình sẽ cố gắng xem cách giải nó thế nào.

Trả lời



myth viết:

Tháng Một 12, 2015 lúc 5:57 chiều

thật sự mình quá lơ mơ về thuật toán của bạn k hiểu được nhiều

Trả lời



nguyenvanquan7826 viết:

Tháng Một 12, 2015 lúc 8:42 chiều

Bạn không hiểu code hay thuật toán? Không hiểu thuật dijkstra hay floyd.

Trả lời



thao viết:

Tháng Một 15, 2015 lúc 4:29 chiều

mình làm thực tập cơ sở về thuật toán này,,,thì giao diện ra nó sẽ như thế nào?

Trả lời



nguyenvanquan7826 viết:

Tháng Một 15, 2015 lúc 5:53 chiều

Cái này là tùy bạn thôi. Bạn có thể tham khảo chương trình của mình tại đây:

<https://cachhoc.net/2014/06/14/java-thuat-toan-mo-phong-thuat-toan-dijkstra-tim-duong-di-ngan-nhat/>

Trả lời



Phuong viết:

Tháng Một 15, 2015 lúc 6:44 chiều

anh ơi có code java ko vậy? C e chẳng biết gì cả

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Một 15, 2015 lúc 8:48 chiều

Học đk java mà không rõ C? nó như nhau, chả khác gì luôn.

[Trả lời](#)



caocuong viết:

Tháng Một 30, 2015 lúc 1:34 sáng

ad ơi, e đang có vấn đề về : tìm đường đi dài nhất trên đồ thị k có chu trình bằng thuật toán PERT .. ad giúp e đc k?

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Một 30, 2015 lúc 10:03 sáng

Cái này anh chưa làm mà giờ cũng chưa có thời gian làm nữa 😊 Em thông cảm nhá.

[Trả lời](#)



dattrinh viết:

Tháng Hai 3, 2015 lúc 2:03 chiều

bạn ơi mình sửa 1 tí ở code của bạn là nhập 2 điểm từ bàn phím và chuyển scanf printf thành cin và cout nhưng ko chạy được, bạn giúp mình tí nhé

```
#include
#include
#include
#define INP "input.txt"
using namespace std;

int main() {
FILE *fi = fopen("input.txt", "r");

int n, a, b, i, sum = 0;
cout<>a;
cout<>b;
fscanf(fi, "%d%d%d", &n);
int G[n][n];
int S[n], Len[n], P[n];
for (i = 0; i < n; i++)
for (int j = 0; j < n; j++) {
fscanf(fi, "%d", &G[i][j]);
sum += G[i][j];
}

for (int i = 0; i < n; i++) {
for (int j = 0; j < n; j++) {
if (i != j && G[i][j] == 0)
G[i][j] = sum;
}
}

a--;
b--;

for (int i = 0; i < n; i++) {
Len[i] = sum;
S[i] = 0;
P[i] = a;
}

Len[a] = 0;
while (S[b] == 0) {
for (i = 0; i < n; i++)
if (!S[i] && Len[i] < sum)
break;
for (int j = 0; j < Len[i]) {
i = j;
```




```

}
}

S[i] = 1;

for (int j = 0; j < n; j++) {
    if (!S[j] && Len[i] + G[i][j] < Len[j]) {
        cout<< " duong di ngan nhat tu dinh "<<a + 1<<" den
        dinh"<<b + 1<<" la "<< Len[b];

        while (i != a) {
            cout<< i + 1<<"<- ";
            i = P[i];
        }
        cout<< a + 1;
    } else {
        cout<<"khong co duong di tu "<<a + 1<<" den "<< b + 1;
    }
}
fclose(fi);

return 0;
}

```

[Trả lời](#)



nguyenvanquan7826 viết:
Tháng Hai 3, 2015 lúc 4:50 chiều

Bạn xem lại 2 dòng này nhá.

cout<>a;

cout<>b;

cin là dùng >>

cout là dùng <<

[Trả lời](#)



dattrinh viết:
Tháng Hai 3, 2015 lúc 5:56 chiều

mình dùng cin>> và cout << đó bạn nhưng ko được

[Trả lời](#)



dattrinh viết:

Tháng Hai 3, 2015 lúc 5:58 chiều

mình chỉ chạy được nhập a và b thôi sau đó chương trình sẽ ko chạy nữa nó áo là stop working

[Trả lời](#)



dattrinh viết:

Tháng Hai 3, 2015 lúc 6:01 chiều

mình chạy thì nó chỉ nhập a và b xong rồi thì nó báo stop working

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Hai 4, 2015 lúc 12:25 sáng

Ở trên mình thấy bạn viết thế này mà

cout<>a;

cout<>b;

fscanf(fi, "%d%d%d", &n);

cout viết <>

khi nhập mỗi n thì chỉ có 1 cái %d thôi chứ, 3 cái lận...

[Trả lời](#)



raven viết:

Tháng Ba 10, 2015 lúc 9:51 sáng

bạn ơi cho mình hỏi ở dòng

" for (int j = 0; j < Len[j])

i = j;

thì !S[j] có ý nghĩa gì?

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Ba 11, 2015 lúc 5:35 chiều

!S[j] tức là điểm j chưa được xét đó.

Trả lời



Mac viết:

Tháng Ba 12, 2015 lúc 8:35 sáng

tại sao bạn lại xuất đường đi là $b \leftarrow \dots \leftarrow a$ sao không xuất ngược lại

Trả lời



nguyenvanquan7826 viết:

Tháng Ba 12, 2015 lúc 11:34 sáng

Tại vì thứ tự đỉnh tìm được nó ngược nên cần làm thế. 😊

Trả lời



Mac viết:

Tháng Ba 14, 2015 lúc 4:07 chiều

mình có thể đổi lại được không



nguyenvanquan7826 viết:

Tháng Ba 15, 2015 lúc 1:49 chiều

Được, bạn xem phần code theo hàm ở bên trên nhé.



Nevermore viết:

Tháng Ba 14, 2015 lúc 4:52 chiều



Anh ơi, em lấy code hoàn chỉnh của anh ở <http://pastebin.com/FiZzb3UH> copy vào devc++ chạy thì nó không ra giao diện như của anh ạ, nó chạy như ma trận ấy. Em dùng Devc++5.7.1 ạ. Anh chỉ em làm sai chỗ nào với!

Trả lời



nguyenvanquan7826 viết:

Tháng Ba 15, 2015 lúc 1:43 chiều

Là sao bạn, cái code này chạy nó sẽ ra giao diện như mấy hình phía cuối bài viết của mình mà? Bạn muốn nó hiện như thế nào?

Trả lời



thuylanh viết:

Tháng Tư 14, 2015 lúc 4:37 chiều

a có thể chuyển code này sang java ko ạ.

Trả lời



nguyenvanquan7826 viết:

Tháng Tư 14, 2015 lúc 11:44 chiều

C với java khác gì đâu bạn. Nếu muốn có thể xem bài này nhé:

<https://cachhoc.net/2014/06/14/java-thuat-toan-mo-phong-thuat-toan-dijkstra-tim-duong-di-ngan-nhat/>

Trả lời



VuDuc viết:

Tháng Tư 19, 2015 lúc 7:53 chiều

b ơi! b có thể giúp mình làm với đồ thị có hướng dc k????

Trả lời



VuDuc viết:

Tháng Tư 19, 2015 lúc 8:58 chiều

nửa trên với nửa dưới của ma trận là 2 hướng luôn đúng k b???

Trả lời



nguyenvanquan7826 viết:

Tháng Tư 20, 2015 lúc 6:04 sáng

Dúng rồi bạn, bạn chỉ cần cho 2 nửa khác nhau là thành đồ thị có hướng mà. VD 1->2 nhưng 2 ko đi đến 1 thì $A[1][2] = 2$, còn $A[2][1] = \text{vô cùng}$.

Trả lời



duyet viết:

Tháng Tư 20, 2015 lúc 2:04 chiều

a có thể làm mô phỏng về thuật toán này không?? giải thích rõ code bằng mô phỏng a?? e mới học lập trình mong a giúp.

Trả lời



nguyenvanquan7826 viết:

Tháng Tư 20, 2015 lúc 9:24 chiều

Bạn có thể xem ở đây nhé

<https://cachhoc.net/2014/06/14/java-thuat-toan-mo-phong-thuat-toan-dijkstra-tim-duong-di-ngan-nhat/>

Trả lời



Veoo viết:

Tháng Tư 28, 2015 lúc 4:51 chiều

Tuyệt ! Rất thích cách trình bày của bạn.

Trả lời



tun viết:

Tháng Tư 29, 2015 lúc 5:46 chiều

bạn ơi cho mình hỏi file input.inp thì tạo thế nào và để ở đâu vậy. mình dùng visual studio 2010 chạy có đc không

Trả lời



nguyenvanquan7826 viết:

Tháng Tư 29, 2015 lúc 6:02 chiều

finle input.inp là 1 file text thôi, để cùng thư mục với file .c hoặc .cpp

VS thì mình không rõ. Mình chưa dùng nó, bạn cứ thử xem

Trả lời



tun viết:

Tháng Năm 1, 2015 lúc 10:30 sáng

ok. mình cảm ơn b nhé :v

Trả lời



cuwowng viết:

Tháng Năm 8, 2015 lúc 8:25 chiều

cho mình hỏi thuật toán warshall này của mình viết kiểu này nó chỉ hiển thị đúng từ w0 đến p6 là kết thúc mà không thấy hiển thị đường đi ngắn nhất từ đâu đến đâu cả . bạn xem và có thể sửa hộ giúp mình được k.
code :

#include

```
#include
int main()
{
    int w[10][10];
    int k, n, i, j;
    int p[10][10];
    printf("nhap vao kich thuong n:"); scanf("%d",&n);
    //nhap vao w[0] va p[0]
    for (i=0; i<n; i++) for (j=0; j<n; j++) {
        printf("w[%d][%d]:", i, j);
        scanf("%d", &w[i][j]);
    }
    for (i=0; i<n; i++) for (j=0; j<n; j++) {
        printf("p[%d][%d]:", i, j);
        scanf("%d", &p[i][j]);
    }
    //tinh toan ra in ra w[k],p[k] voi k=1,2,...,k
    for (k=1; k<n; k++) {
        for (i=0; i<n; i++) for (j=0; j<n; j++) {
            if (p[i][j]>w[i][k]+w[k][j])
                p[i][j]=w[i][k]+w[k][j];
        }
        printf("k=%d\n", k);
    }
    //tim ra duong di ngan nhat
    printf("Nhap dinh xuat phat s = "); scanf("%d",&s);
```



```
printf("Nhap dinh ket thuc f = "); scanf("%d",&f);
Dijkstra(n,a,L,pi,s);
if(L[f]==vc) printf("Khong co duong di");
else
{
printf("\n Duong di tu %d den %d ngan nhat %d\n",s,f,L[f]);
induong(s,f,pi);
}
delete L;
delete pi;
getch();
}
```

Trả lời



nguyenvanquan7826 viết:

Tháng Năm 8, 2015 lúc 10:23 chiều

Cái thuật này mình chưa tìm hiểu nên xin lỗi bạn mình chưa rõ 😊

Trả lời



ducmanhkthd viết:

Tháng Năm 9, 2015 lúc 9:19 chiều

anh cho em hỏi,muốn đánh giá thuật toán Floyd như thế nào ạ? em chỉ biết nó ra $O(n^3)$ anh có thể hướng dẫn em đánh giá thuật toán này ko?thank ạ

Trả lời



nguyenvanquan7826 viết:

Tháng Năm 9, 2015 lúc 9:33 chiều

Bạn xem cách đánh giá ở bài này nhé:

<https://www.cachhoc.net/2013/06/14/thuat-toan-p1->

[cach-tinh-do-phuc-tap-thuat-toan-algorithm-complexity/](#)

Trả lời



cươg viết:

Tháng Năm 10, 2015 lúc 9:15 sáng

cho mình hỏi cái code nâng cao ở trên sao mình chạy trên code block bị lỗi gì ý các chữ chạy loạn lên

Trả lời



nguyenvanquan7826 viết:

Tháng Năm 10, 2015 lúc 10:49 sáng

Mình chạy vẫn ổn mà, nó báo lỗi thế nào bạn?

Trả lời



cươg viết:

Tháng Năm 10, 2015 lúc 1:46 chiều



Trả lời



nguyenvanquan7826 viết:

Tháng Năm 11, 2015 lúc 12:07 sáng

*Do bạn đặt file là *.c nên nó không phải c++ nên không có các thư viện đó. Các thư viện đó là của C++ bạn phải đặt file là *.cpp*

Trả lời



duyet viết:

Tháng Năm 19, 2015 lúc 6:16 chiều

cho e hỏi tại sao floyd dùng được trọng số âm còn dijkstra lại không?? nói rõ e được không?? e k hiểu

Trả lời



nguyenquan7826 viết:

Tháng Năm 20, 2015 lúc 9:31 sáng

Chào bạn, xin lỗi bạn hôm qua khi bạn gọi cho mình và mình đã khẳng định sai lệch khi cho rằng thuật toán vẫn đúng với trọng số âm. Để biết về cái này chúng ta cần dựa vào cách chứng minh thuật toán:

"Chúng ta sẽ chỉ ra, khi một đỉnh v được bổ sung vào tập S , thì $d[v]$ là giá trị của đường đi ngắn nhất từ nguồn s đến v .

Theo định nghĩa nhãn d , $d[v]$ là giá trị của đường đi ngắn nhất trong các đường đi từ nguồn s , qua các đỉnh trong S , rồi theo một cạnh nối trực tiếp $u-v$ đến v .

Giả sử tồn tại một đường đi từ s đến v có giá trị bé hơn $d[v]$. Như vậy trong đường đi, tồn tại đỉnh giữa s và v không thuộc S . Chọn w là đỉnh đầu tiên như vậy.

Đường đi của ta có dạng $s - \dots - w - \dots - v$. Nhưng do trọng số các cạnh không âm nên đoạn $s - \dots - w$ có độ dài không lớn hơn toàn bộ đường đi, và do đó có giá trị bé hơn $d[v]$. Mặt khác, do cách chọn w của ta, nên độ dài của đoạn $s - \dots - w$ chính là $d[w]$. Như vậy $d[w] < d[v]$, trái với cách chọn đỉnh v . Đây là điều mâu thuẫn. Vậy điều giả sử của ta là sai. Ta có điều phải chứng minh." Như vậy nếu trọng số của các cạnh có thể âm thì ta không khẳng định được "đoạn $s - \dots - w$ có độ dài bé hơn đoạn $s-w-v$ " và như vậy ta không thể chứng minh tiếp. Trong khi đó xét thuật toán floyd: "Để đi từ $a \rightarrow b$. Bạn mất 1 quãng đường là x . Thuật toán sẽ tìm 1 đường đi gián tiếp từ $a \rightarrow k \rightarrow b$ và nếu đường đi này ngắn hơn đường đi trực tiếp thì ta gán luôn giá trị nhỏ nhất của đường đi trực tiếp bằng đường đi gián tiếp." Do việc tìm đường đi gián tiếp ngay từng bước

mà không tìm ngắn nhất từng đoạn như Dijkstra nên sẽ không bị ảnh hưởng bởi trọng số âm hay dương.

Trả lời



Tiến viết:

Tháng Năm 26, 2015 lúc 7:08 chiều

Bạn ơi cho mình hỏi là tìm đường đi ngắn nhất khác với tìm đường đi có trọng số nhỏ nhất à. Có cần quan tâm đến trọng số k nhỉ?

Trả lời



nguyenvanquan7826 viết:

Tháng Năm 26, 2015 lúc 11:49 chiều

Về bản chất thì khác nhưng các bài toán chúng ta xét thì hầu như coi trọng số chính là quãng đường đi nên không khác.

Trả lời



Tiến viết:

Tháng Năm 27, 2015 lúc 1:07 sáng

Cho mình hỏi có cần tìm $V+$ – không?

Trả lời



nguyenvanquan7826 viết:

Tháng Năm 27, 2015 lúc 1:09 sáng

$V+-$ là gì bạn?

Trả lời



thường viết:

Tháng Mười Một 1, 2015 lúc 10:22 chiều

bạn ơi bạn có code của thuật toán FORD & FULKERSON
ko cho mình xin

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Mười Một 2, 2015 lúc 12:36 sáng

Mình không có bạn ah.

[Trả lời](#)



tỉnh ngổ viết:

Tháng Năm 10, 2016 lúc 11:42 chiều

bạn có code c cho bài trên cho mình xin vñ mình chưa học c++

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Năm 11, 2016 lúc 12:53 sáng

Bài kia mình code c mà.

[Trả lời](#)



hoang viết:

Tháng Năm 14, 2016 lúc 10:36 chiều

sao e chạy thử code dijkstra phần 1 lại bị lỗi stop working nhĩ?? a xem lại đc ko ạ?

[Trả lời](#)



nguyenvanquan7826 viết:

Tháng Năm 19, 2016 lúc 10:53 chiều

Bạn check xem lỗi gì nhé.

Trả lời



Nguyễn Việt Dũng viết:

Tháng Sáu 3, 2016 lúc 3:08 sáng

Bạn ơi sao của mình nó cứ báo lỗi thế này nhĩ

<http://www.mediafire.com/view/aowfsuq6jpz5a0c/Loi.jpg>

Trả lời



nguyenvanquan7826 viết:

Tháng Sáu 20, 2016 lúc 9:43 chiều

Mình ko code bằng vs c nên ko rõ.

Trả lời



noname viết:

Tháng Tám 12, 2016 lúc 7:24 chiều

Hình như code Java này bạn sưu tầm của tác giả nào đó!

Trả lời



nguyenvanquan7826 viết:

Tháng Tám 17, 2016 lúc 11:28 sáng

Mình đâu có coe java trong bài này bạn? 😊 Và đảm bảo tất cả các code trong bài này và code Demo chương trình đồ họa bằng Java đều do tay mình viết. Nếu bạn thấy ở chỗ khác thì chỉ có họ copy của mình thôi. 😊 😊

Trả lời



ThongDH viết:

Tháng Mười 15, 2016 lúc 12:36 sáng

Chào Quân

Chỗ này code có vấn đề :

```
*G = (int **) malloc((*n) * sizeof(int));
```

Sửa: sizeof *int

```
*G = (int **) malloc((*n) * sizeof(*int));
```

Có thể bạn run trên Os 32 bit nên kết quả vẫn đúng, trên 64bit chắc là có vấn đề.

Mình chỉ đọc qua thôi chứ chưa debug :))

Thân.

Trả lời



nguyenvanquan7826 viết:

Tháng Mười 15, 2016 lúc 1:01 sáng

Ah rồi, mình có thấy lỗi. 😊 cảm ơn bạn nhé.

Trả lời



le tran phuong nhi viết:

Tháng Mười 16, 2016 lúc 12:27 chiều

a có thể mô phỏng thuật toán AT giúp e dc k?

Trả lời



nguyenvanquan7826 viết:

Tháng Mười 16, 2016 lúc 9:14 chiều

AT là thuật toán gì thế?

Trả lời



Tiến Mướp viết:

Tháng Mười Một 4, 2016 lúc 2:30 chiều

Mình vừa review code thì thấy 1 điểm khá là thắc mắc là :

Đoạn code này bạn gán:

```
int n, a, b, i, sum = 0; // ==> sum = 0 ;  
for (int i = 0; i < Len[0] = 0 , Len[1] = 0 .... Len[n-1] =0;  
{  
    Len[a] = 0;  
    // đoạn ở giữa này không có chỗ nào gán lại Len[i] mà i  
    chạy từ 0 đến n-1 ;  
    //==> Len[0] = 0 , Len[1] = 0 .... Len[n-1] =0;  
    for (int j = 0; j < n; j++) { // tính lại do dai cua cac diem  
        chua xet  
        if (!S[j] && Len[i] + G[i][j] < Len[j]) {  
            Len[j] = Len[i] + G[i][j]; // thay doi len  
            ??? cái điều kiện  $Len[i] + G[i][j] < Len[j]$  có bao giờ xảy ra  
            không? khi mà  $0 + k < 0$  ?
```

Trả lời



nguyenvanquan7826 viết:

Tháng Mười Một 18, 2016 lúc 5:19 chiều

Bạn xem lại code nhé, mình gán cho Len ban đầu là sum trong vòng for
 $Len[i] = sum;$

Trả lời



thu viết:

Tháng Mười Một 14, 2016 lúc 7:53 sáng

a thật nhiệt tình.. cảm ơn vì bài viết

Trả lời



Do chi viết:

Tháng Mười Một 17, 2016 lúc 9:47 sáng

Mk dùng code block nhưng sao k chạy đc hàm con du
mk đã khai báo đầu đủ rồi .

Trả lời



nguyenvanquan7826 viết:

Tháng Mười Một 27, 2016 lúc 6:37 chiều

Bạn nhìn xem nó báo lỗi gì nhé, dựa vào đó mới sửa được 😊

Trả lời



Huy Thông viết:

Tháng Mười Một 18, 2016 lúc 3:51 sáng

Hay

Trả lời



Phương viết:

Tháng Năm 1, 2017 lúc 9:47 sáng

sao mình chạy code nó báo là không tìm thấy file input
là sao vậy ad, chỉ e với

Trả lời



nguyenvanquan7826 viết:

Tháng Năm 5, 2017 lúc 12:02 sáng

Bạn nhớ để file input.inp cùng thư mục với file code nhé.

Trả lời



Hải viết:

Tháng Mười Hai 12, 2017 lúc 11:18 chiều

a ơi cho e hỏi, nếu xây dựng đồ thị từ ma trận chứ k đọc trực tiếp từ file có được k a?

Trả lời



nguyenvanquan7826 viết:

Tháng Mười Hai 13, 2017 lúc 11:28 sáng

Được. Đọc từ file cũng là đọc ma trận thôi mà.

Trả lời



Hữu Tín viết:

Tháng Mười Hai 13, 2017 lúc 11:54 sáng

a ơi code chạy ko thể tìm đường đi được a
a có thể gửi code hoàn chỉnh qua e đc ko a
ví dụ đường đi từ 1->3 chưa thể hiển thị kết quả được a
a có thể Demo kết quả e xem thử đc ko a

Trả lời



nguyenvanquan7826 viết:

Tháng Mười Hai 13, 2017 lúc 12:11 chiều

Code mình chạy chuẩn rồi đó. Trong bài mình cũng có ví dụ luôn rồi.

Trả lời



Hữu Tín viết:

Tháng Mười Hai 13, 2017 lúc 10:22 chiều

code chưa chạy đc đường đi ngắn nhất sao a @ ví dụ:
1<-2<-3<-4

Trả lời



Hữu Tín viết:

Tháng Mười Hai 13, 2017 lúc 10:30 chiều

không có chỗ nhập vị trí bắt đầu vị trí kết thúc sao a

Trả lời



nguyenvanquan7826 viết:

Tháng Mười Hai 13, 2017 lúc 10:30 chiều

Chính là ở dòng đầu tiên của đầu vào đó. Em đọc kỹ bài nhé.

Trả lời



Hữu Tín viết:

Tháng Mười Hai 19, 2017 lúc 3:59 chiều

a có thể cho code chạy bằng file.txt đc ko a

Trả lời



nguyenvanquan7826 viết:

Tháng Mười Hai 19, 2017 lúc 8:37 chiều

Thì code này chạy bằng cách nhập xuất file đó bạn.

Trả lời

Trả lời

Nguyenvanquan7826 rất mong muốn các bạn để lại những phản hồi đóng góp ý kiến, thắc mắc cho bài viết cũng như cho Blog. Nhưng mọi phản hồi mang tính chất **nhờ giải hộ bài tập đều bị xóa** thẳng tay và không phản hồi lại!

Bình luận



Tên *

Email *

Trang web

Phản hồi

- ☐ Thông báo cho tôi bằng email khi có bình luận cho mục này
- ☐ Thông báo cho tôi bằng email khi có bài đăng mới

Phản hồi gần đây

- nguyenvanquan7826 trong [Lập trình C: Bài 2 – Kiểu dữ liệu và nhập xuất trong C](#)
- kiều anh trong [Lập trình C: Bài 2 – Kiểu dữ liệu và nhập xuất trong C](#)

Liên kết

[Lập trình & Cuộc sống](#)
[Thiết kế website chuyên nghiệp Hà Nội](#)
[Thiết kế website bán hàng ở Hà Nội](#)
[Nhà Đất Việt Bắc Thái Nguyên](#)
[Tự học lập trình](#)

- nguyenvanquan7826 [Thiết kế bởi Cách Học - Nguồn bởi WordPress](#) trong [\[Thuật toán – C/C++\] Quick Sort – Các vấn đề liên quan](#)

- Nghiia_Tran trong
[Thuật toán – C/C++]
Quick Sort – Các vấn đề
liên quan
- nguyenvanquan7826
trong [Java – Game] Trò
chơi Pikachu
(Pokemon)

