Greedy Algorithms | Set 5 (Prim's Minimum Spanning Tree (MST))

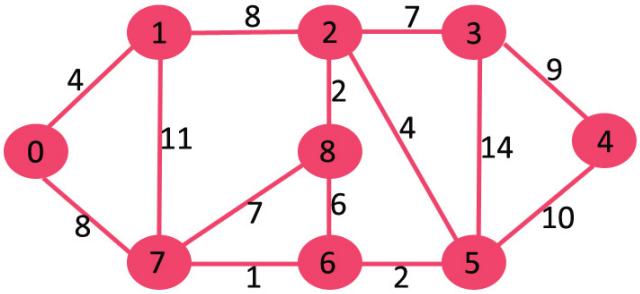
We have discussed Kruskal's algorithm for Minimum Spanning Tree. Like Kruskal's algorithm, Prim's algorithm is also a Greedy algorithm. It starts with an empty spanning tree. The idea is to maintain two sets of vertices. The first set contains the vertices already included in the MST, the other set contains the vertices not yet included. At every step, it considers all the edges that connect the two sets, and picks the minimum weight edge from these edges. After picking the edge, it moves the other endpoint of the edge to the set containing MST.
A group of edges that connects two set of vertices in a graph is called cut in graph theory. *So, at every step of Prim's algorithm, we find a cut (of two sets, one contains the vertices already included in MST and other contains rest of the verices), pick the minimum weight edge from the cut and include this vertex to MST Set (the set that contains already included vertices).*

How does Prim's Algorithm Work? The idea behind Prim's algorithm is simple, a spanning tree means all vertices must be connected. So the two disjoint subsets (discussed above) of vertices must be connected to make a *Spanning* Tree. And they must be connected with the minimum weight edge to make it a *Minimum* Spanning Tree.

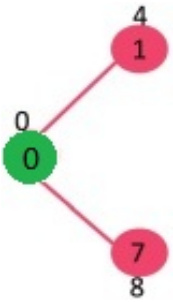
Algorithm
1) Create a set *mstSet* that keeps track of vertices already included in MST.
2) Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign key value as 0 for the first vertex so that it is picked first.
3) While mstSet doesn't include all vertices
….**a)** Pick a vertex *u* which is not there in *mstSet* and has minimum key value.
….**b)** Include *u* to mstSet.
….**c)** Update key value of all adjacent vertices of *u*. To update the key values, iterate through all adjacent vertices. For every adjacent vertex *v*, if weight of edge *u-v* is less than the previous key value of *v*, update the key value as weight of *u-v*

The idea of using key values is to pick the minimum weight edge from cut. The key values are used only for vertices which are not yet included in MST, the key value for these vertices indicate the minimum weight edges connecting them to the set of vertices included in MST.

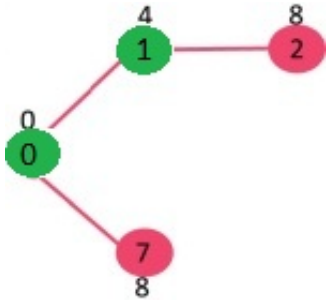
Let us understand with the following example:



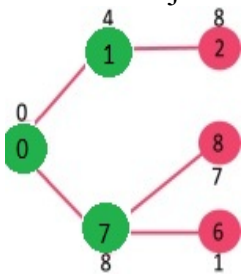
The set *mstSet* is initially empty and keys assigned to vertices are {0, INF, INF, INF, INF, INF, INF, INF} where INF indicates infinite. Now pick the vertex with minimum key value. The vertex 0 is picked, include it in *mstSet*. So *mstSet* becomes {0}. After including to *mstSet*, update key values of adjacent vertices. Adjacent vertices of 0 are 1 and 7. The key values of 1 and 7 are updated as 4 and 8. Following subgraph shows vertices and their key values, only the vertices with finite key values are shown. The vertices included in MST are shown in green color.



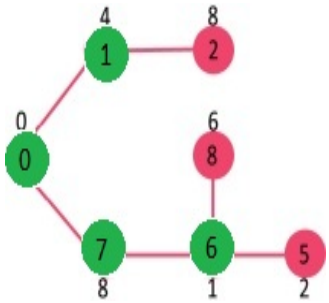
Pick the vertex with minimum key value and not already included in MST (not in mstSET). The vertex 1 is picked and added to mstSet. So mstSet now becomes {0, 1}. Update the key values of adjacent vertices of 1. The key value of vertex 2 becomes 8.



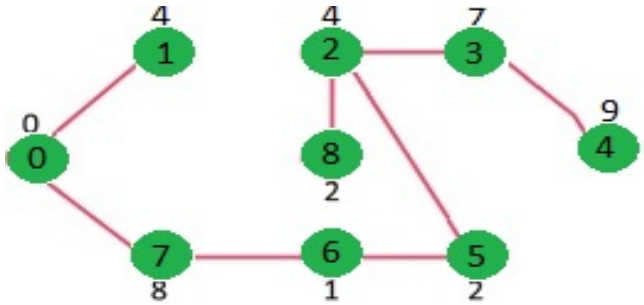
Pick the vertex with minimum key value and not already included in MST (not in mstSET). We can either pick vertex 7 or vertex 2, let vertex 7 is picked. So mstSet now becomes {0, 1, 7}. Update the key values of adjacent vertices of 7. The key value of vertex 6 and 8 becomes finite (7 and 1 respectively).



Pick the vertex with minimum key value and not already included in MST (not in mstSET). Vertex 6 is picked. So mstSet now becomes {0, 1, 7, 6}. Update the key values of adjacent vertices of 6. The key value of vertex 5 and 8 are updated.



We repeat the above steps until *mstSet* includes all vertices of given graph. Finally, we get the following graph.



How to implement the above algorithm?

We use a boolean array mstSet[] to represent the set of vertices included in MST. If a value mstSet[v] is true, then vertex v is included in MST, otherwise not. Array key[] is used to store key values of all vertices. Another array parent[] to store indexes of parent nodes in MST. The parent array is the output array which is used to show the constructed MST.

```
// A C / C++ program for Prim's Minimum Spanning Tree (MST) algorithm.
// The program is for adjacency matrix representation of the graph

#include <stdio.h>
#include <limits.h>

// Number of vertices in the graph
#define V 5

// A utility function to find the vertex with minimum key value, from
// the set of vertices not yet included in MST
int minKey(int key[], bool mstSet[])
{
   // Initialize min value
   int min = INT_MAX, min_index;

   for (int v = 0; v < V; v++)
     if (mstSet[v] == false && key[v] < min)
         min = key[v], min_index = v;

   return min_index;
}

// A utility function to print the constructed MST stored in parent[]
int printMST(int parent[], int n, int graph[V][V])
{
   printf("Edge   Weight\n");
   for (int i = 1; i < V; i++)
      printf("%d - %d    %d \n", parent[i], i, graph[i][parent[i]]);
}

// Function to construct and print MST for a graph represented using adjacenc
// matrix representation
void primMST(int graph[V][V])
{
     int parent[V]; // Array to store constructed MST
     int key[V];   // Key values used to pick minimum weight edge in cut
     bool mstSet[V];  // To represent set of vertices not yet included in MST

     // Initialize all keys as INFINITE
     for (int i = 0; i < V; i++)
        key[i] = INT_MAX, mstSet[i] = false;

     // Always include first 1st vertex in MST.
     key[0] = 0;     // Make key 0 so that this vertex is picked as first ver
     parent[0] = -1; // First node is always root of MST

     // The MST will have V vertices
     for (int count = 0; count < V-1; count++)
     {
        // Pick thd minimum key vertex from the set of vertices
        // not yet included in MST
        int u = minKey(key, mstSet);
```

```
        // Add the picked vertex to the MST Set
        mstSet[u] = true;

        // Update key value and parent index of the adjacent vertices of
        // the picked vertex. Consider only those vertices which are not yet
        // included in MST
        for (int v = 0; v < V; v++)

           // graph[u][v] is non zero only for adjacent vertices of m
           // mstSet[v] is false for vertices not yet included in MST
           // Update the key only if graph[u][v] is smaller than key[v]
          if (graph[u][v] && mstSet[v] == false && graph[u][v] <  key[v])
             parent[v]  = u, key[v] = graph[u][v];
    }

    // print the constructed MST
    printMST(parent, V, graph);
}


// driver program to test above function
int main()
{
   /* Let us create the following graph
          2    3
      (0)--(1)--(2)
       |   / \   |
      6| 8/   \5 |7
       | /     \ |
      (3)-------(4)
            9          */
   int graph[V][V] = {{0, 2, 0, 6, 0},
                      {2, 0, 3, 8, 5},
                      {0, 3, 0, 0, 7},
                      {6, 8, 0, 0, 9},
                      {0, 5, 7, 9, 0},
                     };

    // Print the solution
    primMST(graph);

    return 0;
}
```

Output:

```
Edge   Weight
0 - 1    2
1 - 2    3
0 - 3    6
1 - 4    5
```

Time Complexity of the above program is O(V^2). If the input graph is represented using adjacency list, then the time complexity of Prim's algorithm can be reduced to O(E log V) with the help of binary heap. We will soon be discussing O(E Log V) algorithm as a separate post.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- Assign directions to edges so that the directed graph remains acyclic
- K Centers Problem | Set 1 (Greedy Approximate Algorithm)
- Find the minimum cost to reach destination using a train
- Applications of Breadth First Traversal
- Optimal read list for given number of days
- Print all paths from a given source to a destination
- Minimize Cash Flow among a given set of friends who have borrowed money from each other
- Boggle (Find all possible words in a board of characters)

Tags: Graph, Greedy, Greedy Algorithm, Minimum Spanning Tree, Prim's Algorithm

Writing code in comment? Please use **ideone.com** and share the link here.

Dave • 6 days ago

I'm trying to write a program that outputs to a graphing program (XDot). I can get a graph from user input, but I haven't got the MST part done. I don't have much of a start but would appreciate a push in the right direction:

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
typedef union IPAddress{
unsigned char b[4];
int add;
}IPAddress;

typedef struct Vertex{
IPAddress ip;
char * name;
struct Edge * edgeList;
```

```
}Vertex;

typedef struct Edge{
```

see more

• Reply • Share ›

Andy Toh • 7 days ago

My compile-time solution, whose output agrees with the above run-time solution:

http://ideone.com/RTynvd

• Reply • Share ›

Techie Me • a month ago

Detailed explanation of calculating the running time of this algorithm based on various data structures used for the priority queue implementation.
http://techieme.in/minimum-spa...

• Reply • Share ›

Guest • a month ago

@GeeksforGeeks Diagrammatic representation could be made much better by writing indices outside the nodes and initializing the start node as 0 inside it and INF in all other nodes.

• Reply • Share ›

Guest • 3 months ago

"We repeat the above steps until mstSet doesn't include all vertices of given graph"

Shouldn't this be "We repeat the above steps until mstSet include all vertices of given graph" ?

• Reply • Share ›

GeeksforGeeks Mod ➦ Guest • a month ago

Thanks for pointing this out. We have updated the statement.

• Reply • Share ›

Guest • 3 months ago

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#define max 50
char pre[10];
char sta[10];
int minp[10];
int adj[max][max],W[max][max];
int len[10];
```

```
int max_edge,i,graph_type,wg,j,origin,destin;
int n;//no of vertices(hosts).

void create(){
printf("\n\tEnter the no. of vertices.");
scanf("%d",&n);
max_edge=n*(n-1)/2;

for(i=1;i<=max_edge;i++){
```

see more

• Reply • Share ›

disqus_W3Jq2BTiUC • 3 months ago

// To represent set of vertices not yet included in MST

should read like

// To represent set of vertices included in MST so far

• Reply • Share ›

aprox • 3 months ago

function : int printMST should have void return type

• Reply • Share ›

Viraj • 4 months ago

In the printMST function, shouldn't the weight be graph[parent[i]][i] ?

P.S - Nevermind its a matrix so its fine but it causes a confusion.

• Reply • Share ›

Wander • 5 months ago

There are a ton of errors in this code. For example, INT_MAX is used without declaration.

• Reply • Share ›

Kartik ➚ Wander • 5 months ago

It worked fine for me. Did you compile the program? Which compiler you used?

INT_MAX is defined in limits.h.

• Reply • Share ›

Vivek Kumar • 6 months ago

it doesn't tell all the vertices coming in shortest path between two vertex

• Reply • Share ›

Anurag Singh ➚ Vivek Kumar • 6 months ago

MST and shortest path are two different problems. In MST, we just look for a tree which connects all nodes with minimum cost/weight/length. Program outputs the edges for such a tree.

• Reply • Share ›

Emmanuel Livingstone • 10 months ago

Nice post. Is there another post of the prim's algo that describes the usage heap to reduce time complexity of picking minimum weight crossing edge in the graph cut. I've seen the implementation in Robert Sedgewick's algorithms using Indexed Minimum priority queue. I'd like to see if there is a more sleeker version of it in geeksforgeeks.org

• Reply • Share ›

Emmanuel Livingstone ➚ Emmanuel Livingstone • 9 months ago

I guess my answer was here http://www.geeksforgeeks.org/g...

• Reply • Share ›

sumanth • 10 months ago

please correct the code!!!!

• Reply • Share ›

sumanth ➚ sumanth • 10 months ago

sry!! it's correct

• Reply • Share ›

Guest • a year ago

please mention the line of the program that assigns the parent of the first child of the root

1 • Reply • Share ›

steve • a year ago

hey friends i have a question here. i have a text file as shown below:

A
A-B:3
A-C:4
B-C:2
D-C:1
D-E:2

here are the instructions: the first letter(A) on top can be any character as long as its one of the vertices of the graph and shows our starting node(vertex),from this node we should use prim alg.to calc the MST . then the data below the first row is the edges and their weights and should be dynamic meaning not constant. how do we go about that

• Reply • Share ›

Guest • 2 years ago

Why have you included limits.h ?

1 • Reply • Share ›

GeeksforGeeks Mod ➚ Guest • 2 years ago

For INT_MAX

6 • Reply • Share ›

Guest ➚ GeeksforGeeks • a year ago

You don't need to use this at all. You can use the maximum value stored in the graph itself - it is the last element in preorder and inorder traversal. For postorder you will need minimum number which is the first element so still O(1).

1 • Reply • Share ›

Asif Raza • 2 years ago

I used a little bit different approach for implementing same algorithm.. Its all about how a person think to solve problem. Have look at my C++ code for Prim's Algorithm

http://in.docsity.com/en-docs/...

4 • Reply • Share ›

Rohit Jain • 2 years ago

getting trouble..nt compiling!

1 • Reply • Share ›

racks786 • 2 years ago

```
/* Paste your code here (You may delete these lines if not writing code) */
```

1 • Reply • Share ›

GeeksforGeeks • 2 years ago

Please try saving your file as .cpp. The program may not be compiled by all C compilers, but it must be compiled by CPP compilers. If you still get errors, then please let us know error messages.

• Reply • Share ›

Asheen Richards • 2 years ago

I tried compiling this, at first I got 11 errors. I added the stdbool.h header and it compiled. but nothing happens. as in the program just ends without printing anything or doing anything at all actually. I tried printing sentence at the beginning of main to test it and it didn&#039t even print that.

It seems there is something wrong with the primMST function. I&#039m trying to figure it out but

no luck so far. not sure what the problem is. I&#039m codeblocks on windows by the way.

• Reply • Share ›

GeeksforGeeks • 2 years ago

Could you post some of the errors that you got?

• Reply • Share ›

Amit Arora • 2 years ago

please provide an algorithm not code.

• Reply • Share ›

Alien ➚ Amit Arora • 2 years ago

Algorithm is already provided..

Algorithm
1) Create a set mstSet that keeps track of vertices already included in MST.
2) Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign key value as 0 for the first vertex so that it is picked first.
3) While mstSet doesn't include all vertices
….a) Pick a vertex u which is not there in mstSet and has minimum key value.
….b) Include u to mstSet.
….c) Update key value of all adjacent vertices of u. To update the key values, iterate through all adjacent vertices. For every adjacent vertex v, if weight of edge u-v is less than the previous key value of v, update the key value as weight of u-v

3 • Reply • Share ›

Syeda Shehwar • 2 years ago

not getting this code:-(... it is difficult....

• Reply • Share ›

Sal Alturaigi • 2 years ago

I am having trouble getting this code to compile properly. I get numerous errors. Was there some extra compile command you used?

• Reply • Share ›

Srinivas Giduthuri • 2 years ago

Why can't I pick vertex 2 instead of vertex 7 in step 3?

• Reply • Share ›

GeeksforGeeks ➚ Srinivas Giduthuri • 2 years ago

You can pick 2 also. When there are edges of same wights, there may be more than one solutions (MSTs)

• Reply • Share ›

cooldude ➚ GeeksforGeeks • 2 years ago

But if we pick 2 then mst net weight is 41 as compared to picking up7(weight=37) so will we have to take both cases nd compare

```
/* Paste your code here (You may delete these lines if not writing code) */
```

1 ▲ ▼ • Reply • Share ›

GeeksforGeeks ➚ cooldude • 2 years ago

We should same weight minimum spannning tree (trees may be different, but weight must be same) by picking any of the same weight edges. You must be missing something in your calculations.

▲ ▼ • Reply • Share ›

Sanghwa Jung ➚ GeeksforGeeks • 4 months ago

you mean comparing multicase is useless???

▲ ▼ • Reply • Share ›

Murat Gurses • 2 years ago

Sandeep Jain now it works perfectly fine. thanks mate.

▲ ▼ • Reply • Share ›

GeeksforGeeks • 2 years ago

Yes, either put declatations first or compile it as a C++ program

▲ ▼ • Reply • Share ›

Murat Gurses • 2 years ago

I don&#039t think this would compile because you should declare the for loop like this.

int count;
for(count=0; count<10; i++); count should be outside the scope of for loop.

▲ ▼ • Reply • Share ›

Sandeep Jain • 2 years ago

Could you post some of the errors here? See http://ideone.com/lVLNoP for a sample run.

▲ ▼ • Reply • Share ›

Murat Gurses • 2 years ago

is this code working correctly? when I compile it using codeblocks on mac, it gives me several errors.

▲ ▼ • Reply • Share ›

Murat Gurses • 2 years ago

Hi, Is this code working? because when I compile using codeblocks it gives me several errors

Hi. Is this code working? because when I compile using codeblocks it gives me several errors.

• Reply • Share ›

Sisay Hunde • 2 years ago

yes when I runn it gives one error the compiler said it, please correct it and send for me!

• Reply • Share ›

GeeksforGeeks • 2 years ago

Could you please provide details of the error?

• Reply • Share ›

Kunal Chopra ➚ GeeksforGeeks • a year ago

I am getting error ::) expected at line 6..
what to do?

• Reply • Share ›

Momo ➚ GeeksforGeeks • a year ago

For the first time " minKey" should be called after the update operation for the first node adjacent vertices ,because it will return wrong answer for the condition "key[v] <= min" (if written in place of "key[v] < min").

• Reply • Share ›

Sisay Hunde • 2 years ago

the code has some error please correct it code.

• Reply • Share ›

- Interview Experiences
- Advanced Data Structures
- Dynamic Programming
- Greedy Algorithms
- Backtracking
- Pattern Searching
- Divide & Conquer
- Mathematical Algorithms
- Recursion
- Geometric Algorithms

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding "extern" keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

Follow @GeeksforGeeks

Recent Comments

- lt_k

i need help for coding this function in java...

Java Programming Language · 1 hour ago

- Piyush

  What is the purpose of else if (recStack[*i])...

  Detect Cycle in a Directed Graph · 1 hour ago

- Andy Toh

  My compile-time solution, which agrees with the...

  Dynamic Programming | Set 16 (Floyd Warshall Algorithm) · 1 hour ago

- lucy

  because we first fill zero in first col and...

  Dynamic Programming | Set 29 (Longest Common Substring) · 2 hours ago

- lucy

  @GeeksforGeeks i don't n know what is this long...

  Dynamic Programming | Set 28 (Minimum insertions to form a palindrome) · 2 hours ago

- manish

  Because TAN is not a subsequence of RANT. ANT...

  Given two strings, find if first string is a subsequence of second · 2 hours ago