

PRACTICAL NUMBER – 07

NAME – RANVEER M. BHORTEKAR

SECTION/BATCH – A4/B1

ROLL NO. – 06

SUBJECT – DAA

Aim - Implement Hamiltonian Cycle using Backtracking –

Problem Statement - The Smart City Transportation Department is designing a night-patrol route for security vehicles. Each area of the city is represented as a vertex in a graph, and a road between two area is represented as an edge. The goal is to find a route that starts from the main headquarters (Area A), visits each area exactly once, and returns back to the headquarters — forming a Hamiltonian Cycle. If such a route is not possible, display a suitable message.

CODE IN TEXT FORMAT –

```
def next_value(k,n,G,x):  
    while True:  
        x[k]=(x[k]+1)%n  
        if x[k]==0:return  
        if G[x[k-1]][x[k]]==1:  
            j=0  
            while j<k:  
                if x[j]==x[k]:break  
                j+=1  
            if j==k:  
                if (k<n-1) or (k==n-1 and G[x[n-1]][x[0]]==1):  
                    return
```

```

def hamiltonian(k,n,G,x):
    while True:
        next_value(k,n,G,x)
        if x[k]==0:return
        if k==n-1:
            print([chr(65+v) for v in x]+[chr(65+x[0])])
        else:
            hamiltonian(k+1,n,G,x)
G=[
    [0,1,1,0,1],
    [1,0,1,1,0],
    [1,1,0,1,0],
    [0,1,1,0,1],
    [1,0,0,1,0]
]
n=len(G)
x=[0]*n
x[0]=0
hamiltonian(1,n,G,x)

```

CODE SCREENSHOT –

colab.research.google.com/drive/1s4ptrHb7H9eBC42eM60uKrmzpdNNM_s4#scrollTo=a4fki42GbT0y

RANVEER B. A4_B1_06 DAA PRAC 07.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

RAM Disk

```
[7] def next_value(k,n,G,x):
    while True:
        x[k]=(x[k]+1)%n
        if x[k]==0:return
        if G[x[k-1]][x[k]]==1:
            j=0
            while j<k:
                if x[j]==x[k]:break
                j+=1
            if j==k:
                if (k<n-1) or (k==n-1 and G[x[n-1]][x[0]]==1):
                    return
def hamiltonian(k,n,G,x):
    while True:
        next_value(k,n,G,x)
        if x[k]==0:return
        if k==n-1:
            print([chr(65+v) for v in x]+[chr(65+x[0])])
        else:
            hamiltonian(k+1,n,G,x)

G=[
    [0,1,1,0,1],
    [1,0,1,1,0],
    [1,1,0,1,0],
    [0,1,1,0,1],
    [1,0,0,1,0]
]

n=len(G)
x=[0]*n
x[0]=0
hamiltonian(1,n,G,x)
```

```
[ 'A', 'B', 'C', 'D', 'E', 'A' ]
[ 'A', 'C', 'B', 'D', 'E', 'A' ]
[ 'A', 'E', 'D', 'B', 'C', 'A' ]
[ 'A', 'E', 'D', 'C', 'B', 'A' ]
```

Variables Terminal

10:52 PM Python 3