Assignment2_Part2_Final

September 27, 2024

1 Assignment 2

1.1 Longest Closed Path Problem

By: Dattaraj Salunkhe - 22B1296 | Swayam Patel - 22B1816 Group 25

[1]:
```
from pyomo.environ import *
```

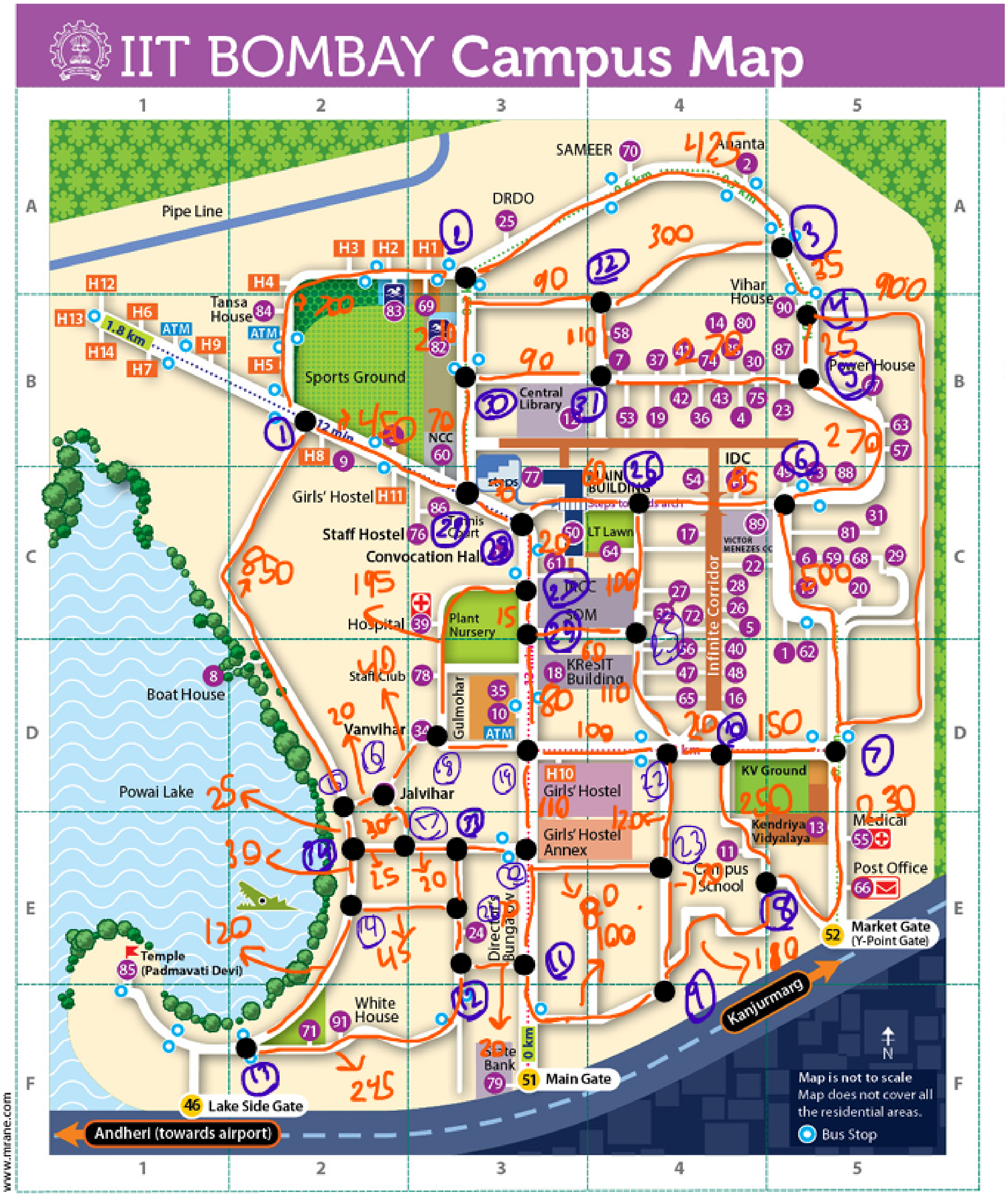
1.1.1 Defining the Graph

We have approximated the given map by excluding the very small white roads. The nodes in the map are landmarks, buildings and some juctions are approximated. We have included a picture of the graph for convinience.

[2]:
```
V = ['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']  # Vertices in the graph

E = [
    ('1', '2'), ('2', '1'),
    ('2', '3'), ('3', '2'),
    ('3', '4'), ('4', '3'),
    ('4', '5'), ('5', '4'),
    ('4', '7'), ('7', '4'),
    ('5', '6'), ('6', '5'),
    ('6', '7'), ('7', '6'),
    ('7', '8'), ('8', '7'),
    ('8', '9'), ('9', '8'),
    ('8', '10'), ('10', '8'),
    ('9', '11'), ('11', '9'),
    ('11', '12'), ('12', '11'),
    ('12', '13'), ('13', '12'),
    ('13', '14'), ('14', '13'),
    ('14', '34'), ('34', '14'),
    ('34', '15'), ('15', '34'),
    ('15', '1'), ('1', '15'),
    ('1', '29'), ('29', '1'),
    ('29', '28'), ('28', '29'),
    ('28', '26'), ('26', '28'),
    ('26', '6'), ('6', '26'),
    ('29', '30'), ('30', '29'),
    ('30', '2'), ('2', '30'),
    ('2', '32'), ('32', '2'),
    ('32', '3'), ('3', '32'),
    ('30', '31'), ('31', '30'),
    ('31', '5'), ('5', '31'),
    ('31', '32'), ('32', '31'),
    ('14', '21'), ('21', '14'),
    ('21', '12'), ('12', '21'),
    ('21', '33'), ('33', '21'),
    ('34', '17'), ('17', '34'),
    ('17', '33'), ('33', '17'),
    ('33', '20'), ('20', '33'),
    ('20', '11'), ('11', '20'),
    ('17', '16'), ('16', '17'),
    ('16', '15'), ('15', '16'),
    ('16', '18'), ('18', '16'),
    ('18', '19'), ('19', '18'),
    ('18', '27'), ('27', '18'),
    ('20', '19'), ('19', '20'),
    ('19', '22'), ('22', '19'),
    ('22', '10'), ('10', '22'),
    ('22', '23'), ('23', '22'),
    ('23', '9'), ('9', '23'),
    ('20', '23'), ('23', '20'),
    ('27', '24'), ('24', '27'),
    ('27', '28'), ('28', '27'),
    ('24', '19'), ('19', '24'),
    ('24', '25'), ('25', '24'),
    ('25', '26'), ('26', '25')
]

edge_lengths = {
    ('1', '2'): 700, ('2', '1'): 700,
    ('2', '3'): 425, ('3', '2'): 425,
    ('3', '4'): 35, ('4', '3'): 35,
    ('4', '5'): 25, ('5', '4'): 25,
    ('4', '7'): 900, ('7', '4'): 900,
    ('5', '6'): 270, ('6', '5'): 270,
    ('6', '7'): 500, ('7', '6'): 500,
    ('7', '8'): 230, ('8', '7'): 230,
    ('8', '9'): 180, ('9', '8'): 180,
    ('8', '10'): 250, ('10', '8'): 250,
    ('9', '11'): 100, ('11', '9'): 100,
    ('11', '12'): 20, ('12', '11'): 20,
    ('12', '13'): 245, ('13', '12'): 245,
    ('13', '14'): 120, ('14', '13'): 120,
    ('14', '34'): 30, ('34', '14'): 30,
    ('34', '15'): 25, ('15', '34'): 25,
    ('15', '1'): 850, ('1', '15'): 850,
    ('1', '29'): 450, ('29', '1'): 450,
    ('29', '28'): 30, ('28', '29'): 30,
    ('28', '26'): 60, ('26', '28'): 60,
    ('26', '6'): 85, ('6', '26'): 85,
    ('29', '30'): 70, ('30', '29'): 70,
    ('30', '2'): 210, ('2', '30'): 210,
    ('2', '32'): 90, ('32', '2'): 90,
    ('32', '3'): 300, ('3', '32'): 300,
    ('30', '31'): 90, ('31', '30'): 90,
    ('31', '5'): 270, ('5', '31'): 270,
    ('31', '32'): 110, ('32', '31'): 110,
    ('14', '21'): 45, ('21', '14'): 45,
    ('21', '12'): 30, ('12', '21'): 30,
    ('21', '33'): 40, ('33', '21'): 40,
    ('34', '17'): 25, ('17', '34'): 25,
    ('17', '33'): 20, ('33', '17'): 20,
    ('33', '20'): 23, ('20', '33'): 23,
    ('20', '11'): 70, ('11', '20'): 70,
    ('17', '16'): 30, ('16', '17'): 30,
    ('16', '15'): 20, ('15', '16'): 20,
    ('16', '18'): 40, ('18', '16'): 40,
    ('18', '19'): 17, ('19', '18'): 17,
    ('18', '27'): 195, ('27', '18'): 195,
    ('20', '19'): 110, ('19', '20'): 110,
    ('19', '22'): 100, ('22', '19'): 100,
    ('22', '10'): 20, ('10', '22'): 20,
    ('22', '23'): 120, ('23', '22'): 120,
    ('23', '9'): 70, ('9', '23'): 70,
    ('20', '23'): 80, ('23', '20'): 80,
    ('27', '24'): 15, ('24', '27'): 15,
    ('27', '28'): 20, ('28', '27'): 20,
    ('24', '19'): 80, ('19', '24'): 80,
    ('24', '25'): 60, ('25', '24'): 60,
    ('25', '26'): 100, ('26', '25'): 100
}
```



1.1.2 Creating a model in Pyomo and defining decision variables

[3]:
```
# Create a Pyomo model
model = ConcreteModel()

model.V = Set(initialize=V)
model.E = Set(initialize=E, dimen=2)

# Parameters
```

```
model.l = Param(model.E, initialize=edge_lengths)

# Decision variables
model.x = Var(model.E, within=Binary)  # Binary variable for edge usage
model.y = Var(model.V, within=Binary)  # Binary variable for vertex usage
model.u = Var(model.V, bounds=(1, len(V)), within=NonNegativeReals)  # Subtour elimination using MTZ variables
```

1.1.3 Explanation

- **Edge Selection Variable (x[i,j])**:
  This binary variable indicates whether the edge (i,j) belongs to E is part of the solution.

$$x[i,j] = \begin{cases} 1 & \text{if edge } (i,j) \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

- **Vertex Selection Variable (y[i])**:
  This binary variable indicates whether vertex i belongs to V is included in the cycle.

$$y[i] = \begin{cases} 1 & \text{if vertex } i \text{ is part of the cycle} \\ 0 & \text{otherwise} \end{cases}$$

- **MTZ Variables (u[i])**:
  These continuous variables are used to prevent subtours. They define an ordering of vertices in the cycle to ensure connectivity and enforce that the selected edges form a single cycle. u[i] is constrained between 1 and |V|, the number of vertices.

1.1.4 Objective Function

[4]:

```
# Objective function: Maximize the total length of the selected edges
def objective_rule(model):
    return sum(model.l[i, j] * model.x[i, j] for i, j in model.E)
model.obj = Objective(rule=objective_rule, sense=maximize)
```

1.1.5 Explanation

- **Objective Function (Maximization of Total Edge Length)**:
  The goal is to maximize the total length of the edges selected in the cycle. Each edge (i, j) has a corresponding length l[i,j], and a binary decision variable x[i,j] indicates whether the edge is included in the solution. The objective function sums the lengths of all selected edges:

$$\text{Maximize} \quad \sum_{(i,j)\in E} l[i,j] \cdot x[i,j]$$

  Here, l[i,j] represents the length of the edge from vertex i to vertex j, and x[i,j] = 1 if the edge (i, j) is selected, and x[i,j] = 0 otherwise. The function seeks to maximize the total sum of the selected edge lengths, finding the longest possible cycle in the graph.

1.1.6 Constraints

[5]:

```python
# Constraints:

# Flow conservation: Inflow must equal outflow for each vertex
def flow_conservation_rule(model, i):
    return sum(model.x[i, j] for j in model.V if (i, j) in model.E) == sum(model.
 ↪x[j, i] for j in model.V if (j, i) in model.E)
model.flow_conservation = Constraint(model.V, rule=flow_conservation_rule)

# Degree constraint: Each vertex in the cycle must have exactly two edges (1 in,␣
 ↪1 out)
def degree_rule(model, i):
    return sum(model.x[i, j] + model.x[j, i] for j in model.V if (i, j) in model.
 ↪E or (j, i) in model.E) == 2 * model.y[i]
model.degree = Constraint(model.V, rule=degree_rule)

# MTZ subtour elimination constraints
def mtz_rule(model, i, j):
    if i != '1' and j != '1' and (i, j) in model.E:
        return model.u[i] - model.u[j] + len(V) * model.x[i, j] <= len(V) - 1
    return Constraint.Skip
model.mtz = Constraint(model.V, model.V, rule=mtz_rule)

# Ensure at least one edge is selected
model.force_edges = Constraint(expr=sum(model.x[i, j] for i, j in model.E) >= 1)
```

1.1.7 Explanation

- **Flow Conservation**:
  This constraint ensures that the inflow equals the outflow for each vertex (i belongs to V). For each vertex i, the sum of flows on the outgoing edges is equal to the sum of flows on the incoming edges:

$$\sum_{j \in V:(i,j) \in E} x[i,j] = \sum_{j \in V:(j,i) \in E} x[j,i] \quad \forall i \in V$$

- **Degree Constraint**:
  Each vertex in the cycle must have exactly two edges—one incoming and one outgoing. This is ensured by requiring the sum of the binary decision variables x[i,j] (indicating edge usage) for all edges connected to vertex i to equal twice the vertex decision variable y[i]:

$$\sum_{j \in V:(i,j) \in E \text{ or } (j,i) \in E} (x[i,j] + x[j,i]) = 2 \cdot y[i] \quad \forall i \in V$$

- **MTZ Subtour Elimination**:
  To prevent disconnected cycles (subtours), the MTZ formulation introduces variables u[i],

which define an order for the vertices in the tour. For each pair of distinct vertices (i, j belongs to V) where there is an edge, the following constraint is imposed:

$$u[i] - u[j] + |V| \cdot x[i, j] \leq |V| - 1 \quad \forall i \neq j, \, (i, j) \in E$$

This ensures that no subtours can form unless the edges between i and j are selected.

- **Force Edge Selection**:
  This constraint ensures that at least one edge is selected in the final solution, preventing a trivial result of no edges being chosen:

$$\sum_{(i,j) \in E} x[i, j] \geq 1$$

1.1.8 Solving and Displaying the Results

[6]:

```python
# Solve the model
solver = SolverFactory('glpk')
result = solver.solve(model, tee=True)
```

```
GLPSOL--GLPK LP/MIP Solver 5.0
Parameter(s) specified in the command line:
 --write /var/folders/b9/cqzf91qd1cb5kkpcpbgg1pmm0000gn/T/tmptbjve51h.glpk.raw
 --wglp /var/folders/b9/cqzf91qd1cb5kkpcpbgg1pmm0000gn/T/tmpd0r3u8ez.glpk.glp
 --cpxlp /var/folders/b9/cqzf91qd1cb5kkpcpbgg1pmm0000gn/T/tmpw42ee7pa.pyomo.lp
Reading problem data from
'/var/folders/b9/cqzf91qd1cb5kkpcpbgg1pmm0000gn/T/tmpw42ee7pa.pyomo.lp'...
/var/folders/b9/cqzf91qd1cb5kkpcpbgg1pmm0000gn/T/tmpw42ee7pa.pyomo.lp:1608:
warning: lower bound of variable 'x2' redefined
/var/folders/b9/cqzf91qd1cb5kkpcpbgg1pmm0000gn/T/tmpw42ee7pa.pyomo.lp:1608:
warning: upper bound of variable 'x2' redefined
165 rows, 169 columns, 832 non-zeros
136 integer variables, all of which are binary
1744 lines were read
Writing problem data to
'/var/folders/b9/cqzf91qd1cb5kkpcpbgg1pmm0000gn/T/tmpd0r3u8ez.glpk.glp'...
1401 lines were written
GLPK Integer Optimizer 5.0
165 rows, 169 columns, 832 non-zeros
136 integer variables, all of which are binary
Preprocessing...
165 rows, 169 columns, 832 non-zeros
136 integer variables, all of which are binary
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  3.400e+01  ratio =  3.400e+01
GM: min|aij| =  8.900e-01  max|aij| =  1.124e+00  ratio =  1.262e+00
EQ: min|aij| =  7.929e-01  max|aij| =  1.000e+00  ratio =  1.261e+00
2N: min|aij| =  5.000e-01  max|aij| =  1.062e+00  ratio =  2.125e+00
```

```
Constructing initial basis...
Size of triangular part is 161
Solving LP relaxation...
GLPK Simplex Optimizer 5.0
165 rows, 169 columns, 832 non-zeros
      0: obj =  -0.000000000e+00 inf =   1.000e+00 (1)
      2: obj =   3.064285714e+02 inf =   0.000e+00 (0)
*    98: obj =   7.672205882e+03 inf =   1.092e-14 (0)
OPTIMAL LP SOLUTION FOUND
Integer optimization begins...
Long-step dual simplex will be used
+    98: mip =     not found yet <=              +inf        (1; 0)
+   535: >>>>>   5.417000000e+03 <=   7.631000000e+03  40.9% (31; 1)
+  9461: >>>>>   5.607000000e+03 <=   7.089000000e+03  26.4% (405; 114)
+ 12535: >>>>>   5.757000000e+03 <=   7.002000000e+03  21.6% (508; 201)
+ 13929: >>>>>   5.982000000e+03 <=   6.954000000e+03  16.2% (493; 328)
+ 20828: >>>>>   5.987000000e+03 <=   6.752000000e+03  12.8% (528; 714)
+ 65515: mip =   5.987000000e+03 <=     tree is empty   0.0% (0; 5569)
INTEGER OPTIMAL SOLUTION FOUND
Time used:   1.5 secs
Memory used: 1.8 Mb (1900297 bytes)
Writing MIP solution to
'/var/folders/b9/cqzf91qd1cb5kkpcpbgg1pmm0000gn/T/tmptbjve51h.glpk.raw'...
343 lines were written
```

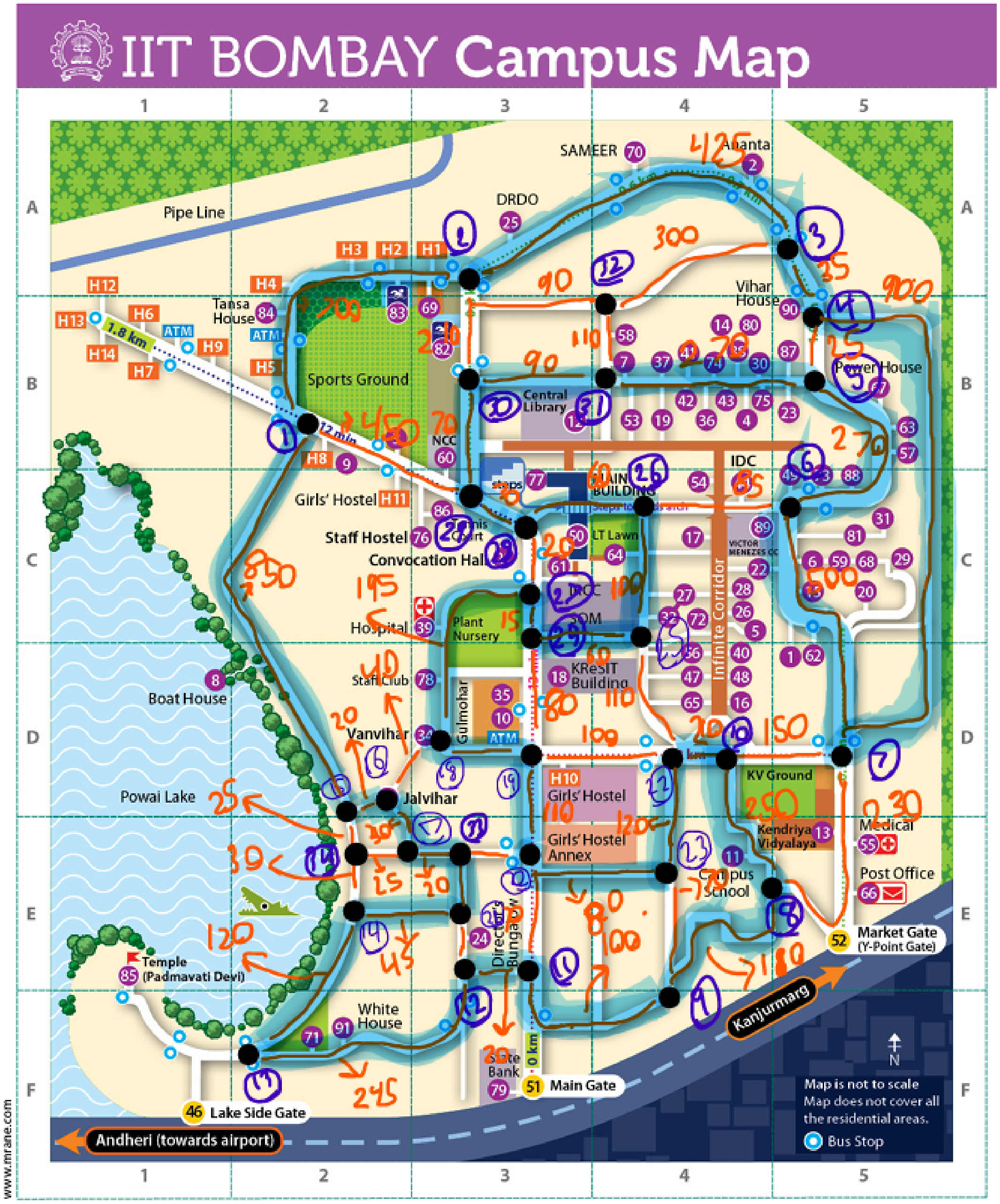
[7]:

```python
# Output the results
print("Status:", result.solver.status)
print("Termination condition:", result.solver.termination_condition)

# Print the selected edges
print("Selected edges in the longest cycle:")
for i, j in model.E:
    if model.x[i, j].value == 1:
        print(f"Edge ({i}, {j}) with length {model.l[i, j]}")

# Print the total length of the longest cycle
print(f"Total length of the longest cycle: {model.obj()}")
```

```
Status: ok
Termination condition: optimal
Selected edges in the longest cycle:
Edge (2, 1) with length 700
Edge (3, 2) with length 425
Edge (4, 3) with length 35
Edge (7, 4) with length 900
Edge (5, 6) with length 270
Edge (6, 7) with length 500
Edge (9, 8) with length 180
Edge (8, 10) with length 250
Edge (11, 9) with length 100
Edge (12, 11) with length 20
Edge (13, 12) with length 245
Edge (14, 13) with length 120
Edge (1, 15) with length 850
Edge (28, 29) with length 30
Edge (26, 28) with length 60
Edge (29, 30) with length 70
Edge (30, 31) with length 90
Edge (31, 5) with length 270
Edge (21, 14) with length 45
Edge (33, 21) with length 40
Edge (17, 33) with length 20
Edge (16, 17) with length 30
Edge (15, 16) with length 20
Edge (19, 18) with length 17
Edge (18, 27) with length 195
Edge (20, 19) with length 110
Edge (10, 22) with length 20
Edge (22, 23) with length 120
Edge (23, 20) with length 80
Edge (27, 24) with length 15
Edge (24, 25) with length 60
Edge (25, 26) with length 100
Total length of the longest cycle: 5987.0
```


IIT BOMBAY Campus Map
1
2
3
4
5
A
B
C
D
E
F
Pipe Line
SAMEER 70
425
Ananta 2
DRDO 25
300
H3
H2
H1
H4
H12
H13
H6
H14
H7
H9
H5
H8 9
ATM
1.8 km
12 min
Tansa House 84
83
69
82
Sports Ground
200
210
450
70
90
90
110
22
Vihar House 90
14 80
58
7
37 41 74 30
270
87
25
900
Power House 67
Central Library 12
31
20
53 19 36 4 23
42 43 75
63
57
270
NCC 60
IDC
Girls' Hostel H11
77
Steps
MAIN BUILDING
Steps to
60
26
54
65
49 3 88
86
Tennis Court
27
Staff Hostel 76
Convocation Hall
28
20
50
LT Lawn
64
61
17
VICTOR MENEZES CC
89
22
31
81
6 59 68 29
500
15
20
850
195
Hospital 39
Plant Nursery
15
IRCC
SOM
100
27
32 72
28
26
5
Infinite Corridor
24
25
56
40
1 62
18 KReSIT Building
47 48
65 16
40
Staff Club 78
35
10
ATM
Gulmohar
80
110
Boat House 8
20
Vanvihar 34
100
200
150
16
7
H10 Girls' Hostel
110
120
Powai Lake
25
Jalvihar
30
17
18
19
21
KV Ground
250
Kendriya Vidyalaya 13
230
Medical 55
Post Office 66
23
11
Campus School
70
30
14
25
20
15
17
80
100
180
8
45
120
24
Directors Bunglow
10
Temple 85 (Padmavati Devi)
9
12
White House 91
71
30
State Bank 79
0 km
51 Main Gate
52 Market Gate (Y-Point Gate)
Kanjurmarg
245
13
46 Lake Side Gate
Andheri (towards airport)
N
Map is not to scale
Map does not cover all the residential areas.
Bus Stop
www.mrane.com