



HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

TỐI ƯU LẬP KẾ HOẠCH



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

TỐI ƯU LẬP KẾ HOẠCH

Mini - Projects

ONE LOVE. ONE FUTURE.

EXAMPLE OF EXPERIMENT REPORTS

data	ILP	CP	Greedy				Hill Climbing				GA				ACO				SA.			
			min	max	avg	std	min	max	avg	std	min	max	avg	std	min	max	avg	std	min	max	avg	std
Ins1	120	120	120	150	135	5.8	120	130	125	3.46	130	150	135	8.4	127	140	135	3.54	125	146	130	13.8
Ins2	120	120	120	130	155	5.8	120	130	125	3.46	130	150	135	8.4	127	140	135	3.54	125	146	130	13.8
Ins3	-	-	160	190	175	5.8	160	190	175	3.46	165	197	185	8.4	167	190	155	3.54	165	196	170	13.8
ins4	-	-	220	250	235	5.8	220	230	225	3.46	230	250	235	8.4	227	240	235	3.54	225	246	230	13.8

Hình 1 Kết quả thí nghiệm về hàm mục tiêu trên các bộ dữ liệu

1. Staff Rostering Problem - Minimize Max night-shift

Có N nhân viên $1, 2, \dots, N$ cần được xếp ca trực làm việc cho các ngày $1, 2, \dots, D$. Mỗi ngày được chia thành 4 kíp: sáng, trưa, chiều, đêm. Biết rằng:

- Mỗi ngày, một nhân viên chỉ làm nhiều nhất 1 ca
- Ngày hôm trước làm ca đêm thì hôm sau được nghỉ
- Mỗi ca trong mỗi ngày có ít nhất A nhân viên và nhiều nhất B nhân viên
- $F(i)$: danh sách các ngày nghỉ phép của nhân viên i

Xây dựng phương án xếp ca trực cho N nhân viên sao cho

- Số ca đêm nhiều nhất phân cho 1 nhân viên nào đó là nhỏ nhất



1. Staff Rostering Problem - Minimize Max night-shift

A solution is represented by a matrix $X[1..N][1..D]$ in which $x[i][d]$ is the shift scheduled to staff i on day d (value 1 means shift morning; value 2 means shift afternoon; value 3 means shift evening; value 4 means shift night; value 0 means day-off)

- **Input**

- Line 1: contains 4 positive integers N, D, A, B ($1 \leq N \leq 500, 1 \leq D \leq 200, 1 \leq A \leq B \leq 500$)
- Line $i + 1$ ($i = 1, 2, \dots, N$): contains a list of positive integers which are the day off of the staff i (days are indexed from 1 to D), terminated by -1

- **Output**

- Line i ($i = 1, 2, \dots, N$): write the i th row of the solution matrix X

1. Staff Rostering Problem - Minimize Max night-shift

- Example

- Input

8 6 1 3

1 -1

3 -1

4 -1

5 -1

2 4 -1

-1

-1

3 -1

- Output

0 1 3 1 4 0

4 0 0 1 2 2

2 4 0 0 2 2

3 1 4 0 0 4

1 0 2 0 1 1

3 2 1 2 3 3

2 3 2 4 0 3

1 3 0 3 1 1

2. People and Parcel Share a Ride

- K taxis (located at point 0) are scheduled to serve transport requests including N passenger requests $1, 2, \dots, N$ and M parcel requests $1, 2, \dots, M$. Passenger request i ($i = 1, \dots, N$) has pickup point i and drop-off point $i + N + M$, and parcel request i ($i = 1, \dots, M$) has pickup point $i + N$ and drop-off point $i + 2N + M$. $d(i, j)$ is the travel distance from point i to point j ($i, j = 0, 1, \dots, 2N + 2M$). Each passenger must be served by a direct trip without interruption (no stopping point between the pickup point and the drop-off point of the passenger in each route). Each taxi k has capacity $Q[k]$ for serving parcel requests. The parcel request i ($i = 1, 2, \dots, M$) has quantity $q[i]$.
- Compute the routes for taxis satisfying above constraints such that the length of the longest route among K taxis is minimal (in order to balance between lengths of taxis).
- A route of a taxi k is represented by a sequence of points visited by that route: $r[0], r[1], \dots, r[Lk]$ in which $r[0] = r[Lk] = 0$ (the depot)

2. People and Parcel Share a Ride

- **Input**

- Line 1: contains N, M , and K ($1 \leq N, M \leq 500, 1 \leq K \leq 100$)
- Line 2: contains $q[1], q[2], \dots, q[M]$ ($1 \leq q[i] \leq 100$)
- Line 3: contains $Q[1], Q[2], \dots, Q[K]$ ($1 \leq Q[i] \leq 200$)
- Line $i + 3$ ($i = 0, 1, \dots, 2N + 2M$): contains the i th row of the distance matrix

- **Output**

- Line 1: contains an integer K
- Line $2k$ ($k = 1, 2, \dots, K$): contains a positive integer Lk
- Line $2k + 1$ ($k = 1, 2, \dots, K$): contains a sequence of Lk integers $r[0], r[1], \dots, r[Lk]$

2. People and Parcel Share a Ride

- Example

- Input

3 3 2

8 4 5

16 16

0 8 7 9 6 5 11 6 11 12 12 12 13

8 0 4 1 2 8 5 13 19 12 4 8 9

7 4 0 3 3 8 4 12 15 8 5 6 7

9 1 3 0 3 9 4 14 19 11 3 7 8

6 2 3 3 0 6 6 11 17 11 6 9 10

5 8 8 9 6 0 12 5 16 15 12 15 15

11 5 4 4 6 12 0 16 18 7 4 3 4

6 13 12 14 11 5 16 0 15 18 17 18 19

11 19 15 19 17 16 18 15 0 13 21 17 17

12 12 8 11 11 15 7 18 13 0 11 5 4

12 4 5 3 6 12 4 17 21 11 0 7 8

12 8 6 7 9 15 3 18 17 5 7 0 1

13 9 7 8 10 15 4 19 17 4 8 1 0

- Output

2

6

0 5 1 7 11 0

10

0 4 6 10 3 9 12 2 8 0

2. People and Parcel Share a Ride

- **Input**

- Line 1: contains N, M , and K ($1 \leq N, M \leq 500, 1 \leq K \leq 100$)
- Line 2: contains $q[1], q[2], \dots, q[M]$ ($1 \leq q[i] \leq 100$)
- Line 3: contains $Q[1], Q[2], \dots, Q[K]$ ($1 \leq Q[i] \leq 200$)
- Line $i + 3$ ($i = 0, 1, \dots, 2N + 2M$): contains the i th row of the distance matrix

- **Output**

- Line 1: contains an integer K
- Line $2k$ ($k = 1, 2, \dots, K$): contains a positive integer Lk
- Line $2k + 1$ ($k = 1, 2, \dots, K$): contains a sequence of Lk integers $r[0], r[1], \dots, r[Lk]$

3. Project Assignment and Scheduling

- A project consists of N tasks $1, 2, \dots, N$ need to be completed by M teams $1, 2, \dots, M$. Team j is available at time point $s(j)$. A task i has a duration $d(i)$ and can be performed by some teams: $c(i, j)$ is the cost if team j performs the task i . Between N tasks, there are precedence constraints represented by Q pairs (i, j) in which task j can only be started to execute after the completion of task i .
- Compute the schedule for performing N tasks (assign a team for each task and specify the starting time-point for each that task) optimizing following criteria (prioritize in the given order):
 1. Number of tasks are scheduled is maximal
 2. The completion time of all tasks is minimal
 3. The total cost for the task assignment is minimal

3. Project Assignment and Scheduling

- **Input**

- Line 1: contains 2 integers N and Q ($1 \leq N, Q \leq 1000$)
- Line $i + 1$ ($i = 1, 2, \dots, Q$): contains i and j in which task j can only be started to execute after the completion of task i
- Line $Q + 2$: contains N positive integers $d(1), d(2), \dots, d(N)$ ($1 \leq d(i) \leq 1000$)
- Line $Q + 3$: contains a positive integer M ($1 \leq M \leq 500$)
- Line $Q + 4$: contains M positive integers $s(1), s(2), \dots, s(M)$ ($1 \leq s(i) \leq 1000$)
- Line $Q + 5$: contains a positive integer K ($1 \leq K \leq 1000000$)
- Line $Q + 5 + k$ ($k = 1, 2, \dots, K$): contains i, j , and $c(i, j)$ in which $c(i, j)$ is the cost when assigning team j to task i

- **Output**

- Line 1: contains a positive integer R
- Line $i + 1$ ($i = 1, \dots, R$): contains 3 positive integer i, j , and u in which task i is assigned to team j and is started to execute at time-point u



3. Project Assignment and Scheduling

- Example

- Input

5 4

1 2

1 4

2 5

3 5

60 45 120 150 20

6

100 20 65 40 25 90

13

1 4 20

1 5 30

1 6 10

2 2 25

2 5 30

3 1 20

3 6 70

4 2 10

4 3 10

4 5 20

5 1 40

5 2 20

5 5 10

- Output

5

1 5 25

2 2 85

3 6 90

4 3 85

5 5 210

4. Order Picking up route in Warehouse

- There are M shelves in a large warehouse $1, 2, \dots, M$ in which the shelf j is located at point j of the warehouse ($j = 1, \dots, M$).
- There are N product $1, 2, \dots, N$. The amount of product i in the shelf j is $Q[i][j]$.
- The warehouse staff starts from the door (point 0) of the warehouse, and want to visit some shelves
- (each shelf is visited at most once) and come back to the door to pickup products for the
- order of customers in which the total amount of product i he must pickup is $q[i]$ ($i = 1, 2, \dots, N$).
- The travel distance from point i to point j is $d(i, j)$ ($0 \leq i, j \leq M$).
- Find the sequence of shelves to visit such that the total travel distance is minimal.

4. Order Picking up route in Warehouse

- **Input**

- Line 1: two positive integer N and M ($1 \leq N \leq 50, 1 \leq M \leq 1000$)
- Line $1 + i$ ($i = 1, \dots, N$): contains the i th row of Q
- Line $N + i + 2$ ($i = 0, 1, \dots, M$): contains the i th row of the distance matrix d
- Line $N + M + 3$: contains $q[1], q[2], \dots, q[N]$

A solution is represented by a sequence of N positive integers x_1, x_2, \dots, x_N representing the sequence of shelves to be visited

- **Output**

- Line 1: contains a positive integer N
- Line 2: contains n positive integers x_1, x_2, \dots, x_N

4. Order Picking up route in Warehouse

- Example

- Input

6 5

3 2 2 4 2

4 3 7 3 5

6 7 2 5 4

2 3 3 2 1

2 5 7 6 1

7 2 1 6 5

0 16 10 13 13 19

16 0 8 3 19 5

10 8 0 7 23 11

13 3 7 0 16 6

13 19 23 16 0 22

19 5 11 6 22 0

8 7 4 8 11 13

- Output

4

2 3 1 5

Explanation: the route of the warehouse staff is: 0 - 2 - 3 - 1 - 5 - 0

5. Balanced Staff Routing for Maintenance

- Có N khách hàng $1, 2, \dots, N$ cần được bảo trì mạng internet. Khách hàng i ở địa điểm i ($i = 1, \dots, N$). Việc bảo trì cho khách hàng i kéo dài $d(i)$ đơn vị thời gian (s). Có K nhân viên kỹ thuật ở trụ sở công ty (điểm 0) và có thời điểm bắt đầu là việc là $t_0 = 0$. Thời gian di chuyển từ điểm i đến điểm j là $t(i, j)$. Lập kế hoạch phân công nhân viên thực hiện bảo trì cho các khách hàng sao cho thời gian làm việc nhiều nhất (thời gian di chuyển cộng thời gian bảo trì) của một nhân viên nào đó là nhỏ nhất
- A route of staff k is represented by a sequence of points $r[0], r[1], r[2], \dots, r[Lk]$ in which $r[0] = r[Lk] = 0$ (the depot)

5. Balanced Staff Routing for Maintenance

- **Input**

- Line 1: contains N and K ($1 \leq N \leq 1000, 1 \leq K \leq 100$)
- Line 2: contains $d(1), d(2), \dots, d(N)$ ($1 \leq d(i) \leq 1000$)
- Line $i + 3$ ($i = 0, 1, 2, \dots, N$): contains the i th row of the matrix t

- **Output**

- Line 1: contains K
- Line $2k$ ($k = 1, \dots, K$): contains a positive integer Lk
- Line $2k + 1$ ($k = 1, 2, \dots, K$): contains $r[0], r[1], r[2], \dots, r[Lk]$



5. Balanced Staff Routing for Maintenance

- Example

- Input

5 2
60 80 70 10 90
0 50 100 60 40 80
50 0 50 40 20 60
100 50 0 50 70 40
60 40 50 0 60 20
40 20 70 60 0 80
80 60 40 20 80 0

- Output

2
5
0 5 3 4 0
4
0 2 1 0

6. Thesis defense Jury Assignment

- Có N đề án tốt nghiệp $1, 2, \dots, N$ và M thầy cô cần được chia vào K hội đồng
 - Mỗi đề án i có $t(i)$ là giáo viên hướng dẫn
 - Giữa 2 đề án i và j có độ tương đồng $s(i, j)$
 - Giữa đề án i và giáo viên j có độ tương đồng $g(i, j)$
- Yêu cầu
 - Số đề án trong mỗi HĐ phải lớn hơn hoặc bằng a và nhỏ hơn hoặc bằng b
 - Số giáo viên trong mỗi HĐ phải lớn hơn hoặc bằng c và nhỏ hơn hoặc bằng d
 - Giáo viên không được ngồi hội đồng của sinh viên mình hướng dẫn
 - Độ tương đồng giữa các đề án trong cùng hội đồng phải lớn hơn hoặc bằng e
- Độ tương đồng giữa đề án với giáo viên trong hội đồng phải lớn hơn hoặc bằng f
- Tổng độ tương đồng giữa các đề án và giữa đề án với giáo viên trong các hội đồng phải lớn nhất
- Mỗi phương án được biểu diễn bởi $x(1), x(2), \dots, x(N)$ và $y(1), y(2), \dots, y(M)$ trong đó $x(i)$ là chỉ số của hội đồng mà đề án i được phân vào, $y(j)$ là chỉ số của hội đồng mà giáo viên j được phân vào.

6. Thesis defense Jury Assignment

- **Input**

- Dòng 1: Ghi N, M và K ($1 \leq N \leq 1000, 1 \leq M \leq 200, 1 \leq K \leq 100$)
- Dòng 2: Ghi a, b, c, d, e, f
- Dòng $2 + i$ ($i = 1, \dots, N$): ghi hàng thứ i của ma trận s
- Dòng thứ $N + 2 + i$ ($i = 1, \dots, N$): ghi hàng thứ i của ma trận g
- Dòng cuối cùng: ghi $t(1), t(2), \dots, t(N)$

- **Output**

- Dòng 1: ghi số nguyên dương N
- Dòng 2; ghi $x(1), x(2), \dots, x(N)$ (các số cách nhau bởi 1 dấu cách SPACE)
- Dòng 3: ghi số nguyên dương M

- Dòng 4 ghi $y(1), y(2), \dots, y(M)$ (các số cách nhau bởi dấu cách SPACE)

6. Thesis defense Jury Assignment

- Example

- Input

6 4 2
2 4 1 3 1 1
0 2 4 1 2 5
2 0 5 5 3 5
4 5 0 4 3 5
1 5 4 0 3 2
2 3 3 3 0 3
5 5 5 2 3 0
3 5 1 5
5 2 5 3

3 1 3 3

5 5 1 3

4 5 4 1

5 3 4 5

1 3 4 2 2 3

- Output

6

2 2 1 1 1 2

4

1 2 1 2

7. CBUS

- There are n passengers $1, 2, \dots, n$. The passenger i want to travel from point i to point $i + n$ ($i = 1, 2, \dots, n$). There is a bus located at point 0 and has k places for transporting the passengers (it means at any time, there are at most k passengers on the bus). You are given the distance matrix c in which $c(i, j)$ is the traveling distance from point i to point j ($i, j = 0, 1, \dots, 2n$). Compute the shortest route for the bus, serving n passengers and coming back to point 0.

7. CBUS

- **Input**

- Line 1 contains n and k ($1 \leq n \leq 1000, 1 \leq k \leq 50$)
- Line $i + 1$ ($i = 1, 2, \dots, 2n + 1$) contains the $(i - 1)^{th}$ line of the matrix c (rows and columns are indexed from $0, 1, 2, \dots, 2n$).

- **Output**

- Line 1: write the value n
- Line 2: Write the sequence of points (pickup and drop-off) of passengers (separated by a SPACE character)

7. CBUS

- Example

- Input

5 3

0 5 8 11 12 8 3 3 7 5 5

5 0 3 5 7 5 3 4 2 2 2

8 3 0 7 8 8 5 7 1 6 5

11 5 7 0 1 5 9 8 6 5 6

12 7 8 1 0 6 10 10 7 7 7

8 5 8 5 6 0 8 5 7 3 4

3 3 5 9 10 8 0 3 4 5 4

3 4 7 8 10 5 3 0 6 2 2

7 2 1 6 7 7 4 6 0 5 4

5 2 6 5 7 3 5 2 5 0 1

5 2 5 6 7 4 4 2 4 1 0

- Output

5

1 2 6 7 5 10 3 4 8 9

8. Class - Course - Teacher Assignment and TimeTabling

- Có T giáo viên 1,2,..., T cần được phân công dạy các môn học cho các lớp.
- Có M môn học 1, 2, ..., M
- Có N lớp học 1, 2,..., N. Mỗi lớp học có 1 danh sách các môn học (lấy từ 1, 2, ..., M). Mỗi lớp học gắn với 1 môn được gọi là lớp-môn
- Mỗi môn học m có số tiết là $d(m)$
- Mỗi giáo viên t có danh sách các môn mà giáo viên đó có thể dạy
- Có 5 ngày học, mỗi ngày chia thành 2 buổi (sáng và chiều), mỗi buổi học được chia thành 6 tiết
- Cần xây dựng kế hoạch phân công giáo viên cũng như thời khóa biểu (ngày/tiết bắt đầu) cho mỗi lớp-môn thỏa mãn:
 - Các lớp-môn của cùng lớp thì không được xếp thời khóa biểu chồng lấp lên nhau
 - Các lớp-môn được phân công cho cùng giáo viên cũng không được xếp thời khóa biểu chồng lấp lên nhau
 - Tổng số lớp-môn được phân kíp và giáo viên là lớn nhất



8. Class - Course - Teacher Assignment and TimeTabling

- **Input**

- Dòng 1: T (số giáo viên), N (số lớp), M (số môn) ($1 \leq N \leq 100, 1 \leq M \leq 100, 1 \leq T \leq 100$)
- Dòng $i + 1$ ($i = 1, \dots, N$): ghi danh sách các môn mà lớp i cần phải học (kết thúc bởi 0)
- Dòng thứ $t + N + 1$ ($t = 1, 2, \dots, T$): ghi danh sách các môn mà giáo viên t có thể dạy (kết thúc bởi 0)
- Dòng thứ $N + T + 2$: ghi $d(m)$ là số tiết của môn m ($m = 1, \dots, M$)

- **Output**

- Dòng 1: ghi số nguyên dương K
- Dòng $k + 1$ ($k = 1, \dots, K$): ghi 4 số nguyên dương x, y, u, v trong đó lớp-môn $x - y$ được phân vào kíp bắt đầu là u và giáo viên dạy là v

8. Class - Course - Teacher Assignment and TimeTabling

- Example

- Input

3 5 4
2 4 0
2 3 4 0
2 3 0
1 2 4 0
1 3 0
1 3 0
2 3 0
1 2 4 0
2 4 4 4

- Output

12
1 2 1 2
1 4 7 3
2 2 1 3
2 3 7 1
2 4 13 3
3 2 7 2
3 3 1 1
4 1 5 1
4 2 13 2
4 4 19 3
5 1 5 3
5 3 13 1

9. Container 2D Loading Minimize Cost

- There are K trucks $1, 2, \dots, K$ to transport N items (2D shape) $1, 2, \dots, N$. Truck k has the container size $W[k] \times L[k]$. Item i has size $w[i] \times l[i]$. Items loaded in a truck can not overlap. The cost of using truck k is $c[k]$. Find a solution that load N items into K trucks such that the total cost of trucks used is minimal.
- A solution is represented by $t[i], x[i], y[i]$, and $o[i]$ in which $(x[i], y[i])$ is the coordinate of item i loaded in truck $t[i]$, $o[i] = 1$, if the item i is rotated 90 degree.



9. Container 2D Loading Minimize Cost

- **Input**

- Line 1: contains N and K ($1 \leq N, K \leq 1000$)
- Line $i + 1$ ($i = 1, 2, \dots, N$) contains 2 integers $w[i]$ and $l[i]$ ($1 \leq w[i], l[i] \leq 1000$)
- Line $1 + N + k$ ($k = 1, \dots, K$) contains $W[k], L[k]$ and $c[k]$ ($1 \leq W[k], L[k] \leq 1000, 1 \leq c[k] \leq 1000$)

- **Output**

- Line i ($i = 1, 2, \dots, N$): write 4 integers $i, t[i], x[i], y[i], o[i]$

9. Container 2D Loading Minimize Cost

- Example

- Input

5 5

90 70

10 70

80 30

100 60

20 90

180 120 8

20 100 10

160 50 6

120 140 11

180 30 7

- Output

1 1 0 0 0

2 1 0 70 1

3 1 0 80 0

4 1 90 0 1

5 1 150 0 0

10. TimeTable assign slot and room to classes

- Có N lớp $1, 2, \dots, N$ cần được xếp thời khóa biểu. Mỗi lớp i có $t(i)$ là số tiết và $g(i)$ là giáo viên đã được phân công dạy lớp đó và $s(i)$ là số sinh viên của lớp. Có M phòng học $1, 2, \dots, M$, trong đó $c(i)$ là số chỗ ngồi của phòng i . Trong tuần có 5 ngày (từ thứ 2 đến thứ 5), mỗi ngày chia thành 12 tiết (6 tiết sáng và 6 tiết chiều). Các tiết của các ngày được đánh số lần lượt từ 1 đến 60.
- Hãy lập thời khóa biểu (xác định ngày, tiết và phòng gán cho mỗi lớp):
 - Hai lớp có chung giáo viên thì phải xếp thời khóa biểu tách rời nhau
 - Số sinh viên trong mỗi lớp phải nhỏ hơn hoặc bằng số chỗ ngồi của phòng học
 - Số lớp được xếp thời khóa biểu là lớn nhất

10. TimeTable assign slot and room to classes

- **Input**

- Line 1: ghi N và M ($1 \leq N \leq 1000, 1 \leq M \leq 100$)
- Line $i + 1$ ($i = 1, \dots, N$): ghi $t(i), g(i)$ và $s(i)$ ($1 \leq t(i) \leq 4, 1 \leq g(i) \leq 100, 1 \leq s(i) \leq 200$)
- Line $N + 2$: ghi $c(1), c(2), \dots, c(M)$ ($1 \leq c(i) \leq 300$)

- **Output**

- Line 1: contains a positive integer Q
- Line $q + 1$ ($q = 1, 2, \dots, Q$): contains 3 positive integers i, u , and v in which class i is assigned to slot u and room v

10. TimeTable assign slot and room to classes

- Example

- Input

10 2
4 1 15
4 1 18
4 1 15
2 2 18
4 2 11
3 1 15
2 2 27
3 2 18
4 1 13
3 1 10
20 20

- Output

9
1 1 1
2 1 2
3 7 1
4 5 1
5 7 2
6 13 1
8 13 2
9 19 1
10 16 1

11. Min Max Vehicle Routing

- Có N điểm $1, 2, \dots, N$ cần thu gom bưu kiện và K bưu tá xuất phát từ bưu điện (điểm 0).
- Biết $d(i, j)$ là khoảng cách từ điểm i đến điểm j , với $i, j = 0, 1, \dots, N$
- Cần xây dựng phương án thu gom cho K bưu tá, xác định mỗi bưu tá thu gom những điểm nào và theo thứ tự nào sao cho quãng đường dài nhất của bưu tá phải ngắn nhất.
- A solution is K routes for K vehicles, each route k is represented by $x[1], x[2], \dots, x[lk]$ in which $x[1] = 0$ (the depot), and $x[2], x[3], \dots, x[lk]$ are pickup points.

11. Min Max Vehicle Routing

- **Input**

- Line 1: contains N and K ($1 \leq N \leq 1000, 1 \leq K \leq 100$)
- Line $i + 1$ ($i = 0, \dots, N$): contains the i th row of the distance matrix d

- **Output**

- Line 1: contains K
- Line $2 * k$ ($k = 1, \dots, K$): contains lk
- Line $2 * k + 1$ ($k = 1, \dots, K$): contains lk integers $x[1], x[2], \dots, x[lk]$ (elements are separated by a SPACE character)

11. Min Max Vehicle Routing

- Example

- Input

```
6 2
0 9 9 9 7 2 9
9 0 3 0 2 8 1
9 3 0 3 4 7 4
9 0 3 0 2 8 1
7 2 4 2 0 6 2
2 8 7 8 6 0 8
9 1 4 1 2 8 0
```

- Output

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

12. TSP with Time Windows

- Một nhân viên giao hàng lấy hàng ở kho (điểm 0) và cần đi giao hàng cho N khách hàng $1, 2, \dots, N$. Khách hàng i nằm ở điểm i và có yêu cầu giao hàng trong khoảng thời gian từ $e(i)$ đến $l(i)$ và giao hàng hết $d(i)$ đơn vị thời gian (s). Biết rằng $t(i, j)$ là thời gian di chuyển từ điểm i đến điểm j . Nhân viên giao hàng xuất phát từ kho tại thời điểm t_0 , hãy tính toán lộ trình giao hàng cho nhân viên giao hàng sao cho tổng thời gian di chuyển là ngắn nhất.

12. TSP with Time Windows

- Each solution is represented by a permutation $s[1], s[2], \dots, s[N]$ of $1, 2, \dots, N$.
- **Input**
 - Line 1: contains a positive integer N ($1 \leq N \leq 1000$)
 - Line $i + 1$ ($i = 1, \dots, N$): contains $e(i), l(i)$ and $d(i)$
 - Line $i + N + 2$ ($i = 0, 1, \dots, N$): contains the i th row of the matrix t .
- **Output**
 - Line 1: contains N
 - Line 2: contains $s[1], s[2], \dots, s[N]$

12. TSP with Time Windows

- **Example**

- **Input**

5

50 90 20

300 350 15

215 235 5

374 404 20

107 147 20

0 50 10 100 70 10

50 0 40 70 20 40

10 40 0 80 60 0

100 70 80 0 70 80

70 20 60 70 0 60

10 40 0 80 60 0

- **Output**

5

1 5 3 2 4

13. Exam TimeTable

- There are N classes $1, 2, \dots, N$ that need to be scheduled for the final exam. Each class must be assigned to a time-slot and a room.
 - There are M rooms $1, 2, \dots, M$ that can be used for scheduling the exam. Each room i has capacity $c(i)$ (number of places of the room)
 - Each day is divided into 4 slots 1, 2, 3, 4.
- Each class i has number of students $d(i)$ ($i = 1, \dots, N$).
- Among N classes, there are K pairs of classes (i, j) in which class i and class j have the same student participating in the exam. It means that these 2 classes cannot be scheduled in the same time-slot.
- **Objective:** Compute the exam time-table such that the number of days used is minimal.

13. Exam TimeTable

A solution is represented by 2 array s and r in which $s[i]$ is the start slot and $r[i]$ is the room of course i

- **Input**

- Line 1: contains N
- Line 2: contains d_1, d_2, \dots, d_N
- Line 3: contains M
- Line 4: contains c_1, c_2, \dots, c_M
- Line 5: contains K
- Line $5 + k$ ($k = 1, \dots, K$): contains 2 integers i and j (2 courses having a same student registerd, these courses cannot be scheduled in the same slot)

- **Output**

- Each line i ($i = 1, 2, \dots, N$): contains 3 integer $i, s[i]$, and $r[i]$

13. Exam TimeTable

- Example

- Input

10 3
72 77 71 71 53 45 53 53
66 70
79 53 70
16
1 2
1 3
1 8
1 10
2 5
2 9

3 6
3 9
4 10
5 8
5 10
7 8
7 9
7 10
8 9
9 10

- Output

1 1 1
2 2 1
3 3 1
4 4 1
5 1 2
6 1 3
7 2 2
8 3 2
9 4 3
10 3 3

14. Paper Reviewer Assignment

- The chair of a conference must assign scientific papers to reviewers in a balance way. There are N papers $1, 2, \dots, N$ and M reviewers $1, 2, \dots, M$. Each paper i has a list $L(i)$ of reviewers who are willing to review that paper. A review plan is an assignment reviewers to papers. The load of a reviewer is the number of papers he/she have to review. Given a constant b , compute the assignment such that:
 - Each paper is reviewed by exactly b reviewers
 - The maximum load of all reviewers is minimal
 - In the solution, each paper i is represented by a list $r(i, 1), r(i, 2), \dots, r(i, b)$ of b reviewers assigned to this paper

14. Paper Reviewer Assignment

- **Input**

- Line 1 contains N, M and b
- Line $i + 1$ ($i = 1, \dots, N$) contains a positive integer k followed by k positive integers representing the list $L(i)$

- **Output**

- Line 1: contains N
- Line $i + 1$ ($i = 1, \dots, N$): contains b and b integers $r(i, 1), r(i, 2), \dots, r(i, b)$ which are the list of reviewers assigned to paper i

14. Paper Reviewer Assignment

- Example

- Input

5 3 2

3 1 2 3

2 1 2

3 1 2 3

3 1 2 3

2 1 3

- Output

5

2 1 2

2 1 2

2 3 1

2 3 2

2 3 1

15. Bin Packing with Lower and Upper Bound Capacity Constraint

- There are N orders $1, 2, \dots, N$, in which order i has quantity $d(i)$ and cost $c(i)$.
- There are K vehicle $1, 2, \dots, K$ for serving orders in which vehicle k low-bound capacity $c_1(k)$ and up-capacity $c_2(k)$. Compute a solution for assigning orders to vehicles such that:
 - Each order is served by at most one vehicle
 - Sum of quantity of orders loaded (served) in a vehicle must be between the low-capacity and up-capacity of that vehicle
 - Total cost of served orders is maximal

15. Bin Packing with Lower and Upper Bound Capacity Constraint

- **Input**

- Line 1: contains positive integers N và K ($1 \leq N \leq 1000, 1 \leq K \leq 100$)
- Line $i + 1$ ($i = 1, \dots, N$): contains 2 integers $d(i)$ and $c(i)$ ($1 \leq d(i), c(i) \leq 100$)
- Line $N + 1 + k$ ($k = 1, \dots, K$): contains 2 integers $c_1(k)$ and $c_2(k)$ ($1 \leq c_1(k) \leq c_2(k) \leq 1000$)

- **Output**

- Line 1: contains an integer m
- Line $i + 1$ ($i = 1, 2, \dots, m$): contains 2 positive integers i and b in which order i is served by vehicle b

15. Bin Packing with Lower and Upper Bound Capacity Constraint

- Example

- Input

5 2

5 9

7 2

12 6

12 4

7 6

12 14

27 31

- Output

5

1 1

2 2

3 2

4 2

5 1

Explanation. orders 1, 5 are served by vehicle 1; order 3, 4 are served by vehicle 2. Order 2 is not served

16. Haverst Planning

- Có N cánh đồng $1 \dots, N$ trồng cùng một loại nông sản
- Mỗi cánh đồng i có sản lượng là $d(i)$ và cần được thu hoạch trong khoảng thời gian từ $s(i)$ đến ngày $e(i)$
- Nhà máy xử lý nông sản có công suất xử lý tối đa là M (tổng sản lượng nông sản có thể xử lý trong 1 ngày). Ngoài ra, khi tổng sản lượng cần xử lý trong một ngày phải lớn hơn hoặc bằng m thì nhà máy mới quyết định mở máy vận hành.
- Hãy lập kế hoạch thu hoạch nông sản (mỗi cánh đồng thu hoạch vào ngày nào) sao cho thỏa mãn ràng buộc về công suất tối đa và tối thiểu của nhà máy đồng thời chênh lệch sản lượng nông sản cần xử lý giữa các ngày sản xuất là nhỏ nhất.

16. Haverst Planning

- **Input**

Line 1: contains 3 positive integers N, m, M

($1 \leq N \leq 10000, 1 \leq m \leq M \leq 10000$)

Line $i+1$ ($i = 1, 2, \dots, N$): contains 3 positive integers $d(i), s(i)$ and $e(i)$

($1 \leq d(i) \leq 100, 1 \leq s(i) \leq e(i) \leq 10000$)

- **Output**

Line 1: contains an integer M which is the number of fields being harvested

Each subsequent line contains i and $x(i)$ (separated by a SPACE character) in case $x[i] > 0$ ($i = 1, 2, \dots, N$)

- **Example**

Input	Output
10 7 24	10
8 3 7	1 3
8 2 6	2 3
4 2 6	3 3
8 1 4	4 1
7 3 7	5 4
7 1 4	6 1
1 1 4	7 3
2 1 4	8 3
6 1 4	9 1
7 1 3	10 2

16. Truck container scheduling

- **Ngữ cảnh bài toán**

Đầu kéo, rơ moóc và container rỗng nằm rải rác ở các bãi (DEPOT) khác nhau. Đầu kéo (TRUCK) cần được lập lộ trình xuất phát từ bãi đỗ, vận chuyển container và kết thúc hành trình tại bãi đỗ. Một TRUCK có thể chở được tối đa 02 container loại 20ft và 01 container loại 40ft. Để vận chuyển container, đầu kéo bắt buộc phải đi lấy rơ moóc (TRAILER) trước và sau khi dùng xong đem trả rơ moóc về bãi. Trong một số trường hợp, TRUCK tới điểm hạ container có thể tháo cả rơ moóc kèm container (do thiếu xe nâng) và đi lấy rơ moóc khác để phục vụ yêu cầu khác. Có 04 loại yêu cầu vận chuyển container:

INBOUND_FULL (IF): chở 1 container đầy từ cảng (PORT) về kho khách hàng (WAREHOUSE) để dỡ hàng nhập khẩu.

INBOUND_EMPTY (IE): chở 1 container rỗng từ WAREHOUSE hoặc PORT về DEPOT để lưu kho.

OUTBOUND_EMPTY (OE): chở 1 container rỗng từ DEPOT tới WAREHOUSE hoặc PORT để bốc hàng xuất khẩu

OUTBOUND_FULL (OF): chở 1 container đầy từ WAREHOUSE tới PORT để xuất khẩu.

- **Các dữ liệu đầu vào:**

Mỗi yêu cầu vận chuyển container gồm các thông tin sau: điểm lấy/trả, loại container (20ft hoặc 40ft), thời gian để lấy/trả container. Thời gian di chuyển từ 1 địa điểm đến 1 địa điểm khác là biết trước. Thời gian tác nghiệp việc lấy và gắn rơ moóc và đầu kéo và thời gian tháo rơ moóc khỏi đầu kéo cũng được biết trước.

16. Truck container scheduling

- **Mục tiêu của bài toán** Cần lập kế hoạch cho các đầu kéo đi lấy rơ-mooc rồi vận chuyển các container theo yêu cầu của khách hàng với mục tiêu hoàn thành công việc (phục vụ hết tất cả các yêu cầu vận chuyển) sớm nhất. Các đầu kéo xuất phát từ bãi của mình, đi lấy rơ mooc ở 1 bãi duy nhất, phục vụ các yêu cầu vận chuyển container và quay về bãi của mình. Thời điểm hoàn thành công việc của mỗi đầu kéo được tính là thời điểm đầu kéo đã phục vụ xong hết các nhiệm vụ vận chuyển được giao và quay trở lại bãi nơi đầu kéo này xuất phát. Hàm mục tiêu của bài toán là thời điểm muộn nhất hoàn thành công việc của một đầu kéo nào đó trong số các đầu kéo được giao thực hiện công việc.

Mục tiêu của bài toán bao gồm 2 thành phần (cần được tối thiểu hóa):

- F1: Thời điểm hoàn thành tất cả yêu cầu vận chuyển là thời điểm hoàn thành công việc muộn nhất trong số thời điểm hoàn thành của các đầu kéo được giao thực hiện công việc.
- F2: Tổng thời gian di chuyển (không tính thời gian phục vụ tháo lắp container, rơ mooc tại các địa điểm) của các đầu kéo.

Hai thành phần F1 và F2 (trong đó F1 được ưu tiên hơn F2) được tổng hợp thành 1 hàm duy nhất

$$F = \alpha * F1 + F2 \text{ (}\alpha \text{ là một hằng số lớn)}$$

16. Truck container scheduling

- **Tiêu chí đánh giá.** Với mỗi testcase, điểm số (score) tùy thuộc vào giá trị của hàm F (hàm F càng nhỏ thì score càng lớn). Điểm số sẽ được xác định bằng tổng điểm số ứng với các testcase.
- Ví dụ có 3 đầu kéo 1, 2, 3 được vận hành để phục vụ các yêu cầu vận chuyển. Thời điểm các đầu kéo 1, 2, 3 hoàn thành công việc và quay về bãi tương ứng là 10000, 12000, 8000. Thời gian di chuyển của đầu kéo 1, 2, 3 tương ứng là 9000, 10000, 6000. Khi đó:

$$F1 = \max(10000, 12000, 8000) = 12000$$

$$F2 = 9000 + 10000 + 6000 = 25000$$

$$F = \alpha * 12000 + 25000$$

Score (điểm) của một phương án được định nghĩa bằng $10000000000 - F$ (score càng lớn thì phương án tìm được càng tốt).

16. Truck container scheduling

- **Tiêu chí đánh giá.** Với mỗi testcase, điểm số (score) tùy thuộc vào giá trị của hàm F (hàm F càng nhỏ thì score càng lớn). Điểm số sẽ được xác định bằng tổng điểm số ứng với các testcase.
- Ví dụ có 3 đầu kéo 1, 2, 3 được vận hành để phục vụ các yêu cầu vận chuyển. Thời điểm các đầu kéo 1, 2, 3 hoàn thành công việc và quay về bãi tương ứng là 10000, 12000, 8000. Thời gian di chuyển của đầu kéo 1, 2, 3 tương ứng là 9000, 10000, 6000. Khi đó:

$$F1 = \max(10000, 12000, 8000) = 12000$$

$$F2 = 9000 + 10000 + 6000 = 25000$$

$$F = \alpha * 12000 + 25000$$

Score (điểm) của một phương án được định nghĩa bằng $10000000000 - F$ (score càng lớn thì phương án tìm được càng tốt).

16. Truck container scheduling

- **Input**

Dòng 1: chứa Points N: trong đó N là số lượng điểm (các địa điểm đánh số từ 1, 2, . . . , N), ($2 \leq N \leq 1000$)

Dòng 2: DISTANCES N^2 :

N^2 dòng tiếp theo, mỗi dòng ghi 3 số nguyên: i j d trong đó d là thời gian di chuyển từ địa điểm i đến địa điểm j ($1 \leq i, j \leq N$)

Dòng tiếp theo ghi: TRAILER p d trong đó p là địa điểm nơi chứa rơ moóc ($1 \leq p \leq N$) và d là thời gian cần để gắn rơ-moóc vào đầu kéo

Dòng tiếp theo ghi: TRUCK m, trong đó m là số đầu kéo ($1 \leq m \leq 500$)

M dòng tiếp theo, mỗi dòng ghi: 2 số nguyên dương i và p trong đó <id> là định danh của đầu kéo (<id> = 1, 2, ..., m) và p là địa điểm nơi đầu kéo <id> đang đỗ

Khối tiếp theo là khối thông tin yêu cầu vận chuyển, khối bao gồm các dòng, mỗi dòng là thông tin của 1 yêu cầu vận chuyển với định dạng:

REQ <id> <size> <p1> <pickup_action> <pickup_duration> <p2> <drop_action> <drop_duration> : trong đó

<id> là định danh của yêu cầu vận chuyển (đánh số 1, 2, ...);

<size> là kích thước container với 2 giá trị là 20 và 40;

<p1>: địa điểm lấy container;

<pickup_action>: là string mô tả hành động lấy container với 2 giá trị PICKUP_CONTAINER: chỉ lấy container, PICKUP_CONTAINER_TRAILER: container đã nằm sẵn trên rơ-moóc rồi, chỉ cần đầu kéo đến vào kéo rơ-moóc chứa container đi;

<pickup_duration>: thời gian diễn ra hoạt động lấy container hoặc gắn rơ-moóc;

<p2>: địa điểm trả container;

<drop_action>: string mô tả hành động trả container với 2 giá trị: DROP_CONTAINER: chỉ hạ container xuống, còn rơ-moóc thì vẫn gắn vào đầu kéo, đầu kéo kéo theo rơ-moóc có thể đi phục vụ yêu cầu vận chuyển khác, DROP_CONTAINER_TRAILER: tháo rơ-moóc khỏi đầu kéo (địa điểm nhận container không có thiết bị hạ container, nên phải tháo và để lại rơ-moóc, container vẫn nằm trên rơ-moóc để bốc dỡ hoặc nạp hàng);

<drop_duration>: khoảng thời gian diễn ra hoạt động trả container.

- Khối thông tin yêu cầu vận chuyển được kết thúc bởi #
- Số yêu cầu vận chuyển có thể lên tới 1000.



16. Truck container scheduling

- **Output**

Dòng đầu tiên ghi ROUTES m, trong đó m là số lượng đầu kéo được phân lộ trình thực hiện phục vụ yêu cầu vận chuyển

Tiếp theo là m khối, mỗi khối chứa thông tin về lộ trình bao gồm chuỗi các địa điểm và hành động tương ứng, mỗi địa điểm và hành động tại địa điểm đó với định dạng:

<point_id> <action> <request_id> trong đó

<point_id> là số thứ tự địa điểm (từ 1 đến N),

<action> là trong mô tả hành động tại địa điểm đó bao gồm các giá trị:

PICKUP_CONTAINER_TRAILER: gắn rơ-mooc có container nằm trên vào đầu kéo

PICKUP_CONTAINER: nhận container (điều kiện đầu kéo đã có sẵn rơ-mooc rồi)

DROP_CONTAINER: hạ container (rơ-mooc vẫn gắn vào đầu kéo)

DROP_CONTAINER_TRAILER: tháo rơ-mooc kèm container ra khỏi đầu kéo

STOP: dừng tại bãi đậu tương ứng của đầu kéo. Với <action> là STOP thì sau đó sẽ không có <request_id>

<request_id>: số nguyên thể hiện định dạng của yêu cầu vận chuyển

Mỗi khối được kết thúc bởi ký tự #



A graphic on the left side of the slide. It features a dark blue background with a large, stylized circular shape composed of many small red dots. The dots are arranged in a way that creates a sense of depth and movement, with some dots appearing larger and more concentrated than others. The word "HUST" is written in white, bold, sans-serif capital letters across the center of this graphic.

HUST

THANK YOU !