Bài Toán 1: Rice (Bờm chuyển gạo).

Bờm là một công nhân của một công ty tinh chế và chế biến lương thực, hôm nay công ty có nhập khẩu *N* bao gạo, đánh số từ *1* đến *N*, bao gạo thứ *i* có khối lượng là M_i gam (M_i nguyên dương). Bờm phải vận chuyển *N* bao gạo này cất vào trong kho theo một trình tự nào đó. Theo yêu cầu của giám đốc, các bao gạo sau khi đưa vào trong kho sẽ được san sang nhau sao cho bao nào cũng có khối lượng là một số nguyên đồng thời không có hai bao nào chênh lệch nhau quá *1* gam.

Bờm quá "thông minh" đến mức sau khi chuyển thêm được một bao gạo nào vào trong kho Bờm lại thực hiện việc san gạo giữa các bao ở trong kho luôn (theo yêu cầu như ở trên) rồi mới ra vận chuyển bao khác. Gọi việc san *1* gam gạo từ bao này sang bao khác có chi phí bằng *1*.

Hãy giúp bờm đưa ra một thứ tự vận chuyển các bao gạo để tổng chi phí san gạo là nhỏ nhất.

Dữ liệu vào: Từ tệp Rice.inp có cấu trúc như sau:

- Dòng 1: Chứa số nguyên dương N $(N<=20)$.

- Dòng 2: Chứa N số nguyên dương $M_1, M_2,..,M_N$ $(M_i \leq 10^9)$.

Kết quả: Ghi ra tệp Rice.out gồm một số nguyên duy nhất là chi phí nhỏ nhất tìm được.

Ví dụ:

Rice.inp	Rice.out
4 1 2 3 4	1

Thuật toán: **Mục tiêu**

Chọn thứ tự đưa N bao gạo (khối lượng M_i) vào kho. Sau mỗi lần thêm 1 bao, kho được san bằng sao cho:

- mọi khối lượng là số nguyên;
- không có hai bao chênh > 1 gam.

  Chi phí = tổng số gam phải san (1 gam = 1 đơn vị chi phí).

  Code đang quy hoạch động trên tập con (bitmask) để **tối thiểu hóa chi phí ước lượng mỗi bước**.

Biểu diễn & tiền xử lý

- Trạng thái x (0…2^n-1) là tập bao đã có trong kho (bit = 1 là đã chọn).
- sobit1[x] = số bao trong trạng thái x.
- S[x] = tổng khối lượng các bao trong x.

2) Ý tưởng tổng quát của code

Code dùng **quy hoạch động trên tập con** (bitmask DP):

- **Trạng thái x**: tập các bao đã đưa vào kho (bit = 1 nghĩa là bao đó đã có).
- **F[x]**: chi phí nhỏ nhất để đạt trạng thái x **và sau đó kho đã được san bằng**.
- Từ x, ta thử **thêm** một bao p chưa có → trạng thái mới y = x ∪ {p}.

  Chi phí tăng thêm được **ước lượng** bởi hàm chiphi(y, p).
- Cập nhật: F[y] = min(F[y], F[x] + chiphi(y,p)).

Bước 1

- Một trạng thái (mask) `x` biểu diễn **tập các bao đã đưa vào kho**:
  - bit $p-1$ = 1 ⇔ bao p đã có trong kho.
  - `sobit1[x]` = số bao hiện có trong kho (số bit 1 của `x`).
  - `S[x]` = tổng khối lượng các bao trong `x`.
- DP:
  - `F[x]` = chi phí nhỏ nhất để đưa **đúng tập** `x` vào kho **và kho đang cân bằng**.
  - Từ `x`, chọn thêm bao **chưa có** `p` → `y = x | (1<<(p-1))`.
  - Cập nhật: `F[y] = min(F[y], F[x] + chiphi(y,p))`.
- Công thức `chiphi(y,p)` (đúng như code):
  - Gọi `k = sobit1[y]`, `S = S[y]`, `tb = floor(S/k)`, `r = S % k`.
  - Nếu `tb < M[p]`:
    - Nếu `r == 0` ⇒ cost = `M[p] - tb` (mọi bao đích = `tb`)
    - Nếu `r > 0` ⇒ cost = `M[p] - (tb+1)` (đích sẽ có `r` bao = `tb+1`)
  - Ngược lại ⇒ cost = `tb - M[p]`.

2) Tiền xử lý `S[x]` và `sobit1[x]` (minh họa nhanh)

Với `M=[1,2,3,4]`:

mask (nhị phân)	tập bao	`sobit1`	`S`
0000	{}	0	0
0001	{1}	1	1
0010	{2}	1	2
0100	{3}	1	3
1000	{4}	1	4
0011	{1,2}	2	3
0101	{1,3}	2	4
0110	{2,3}	2	5
1001	{1,4}	2	5
1010	{2,4}	2	6
1100	{3,4}	2	7
0111	{1,2,3}	3	6
1011	{1,2,4}	3	7
1101	{1,3,4}	3	8
1110	{2,3,4}	3	9
1111	{1,2,3,4}	4	10

3) Khởi tạo DP

- F[0] = 0
- Các F[x>0] = +∞ (trong code là n*OO)

4) Tạo các trạng thái 1 phần tử (từ x=0)

Từ x=0000, thử thêm từng p:

- Thêm p=1 → y=0001:
  - k=1, S=1, tb=1, r=0
  - so với M[1]=1: tb >= M[1] ⇒ cost = tb - M[1] = 0
  - F[{1}] = 0
- Thêm p=2 → y=0010:
  - k=1, S=2, tb=2, r=0, M[2]=2 ⇒ cost = 0 ⇒ F[{2}]=0
- Thêm p=3 → y=0100:
  - k=1, S=3, tb=3, r=0, M[3]=3 ⇒ cost = 0 ⇒ F[{3}]=0
- Thêm p=4 → y=1000:
  - k=1, S=4, tb=4, r=0, M[4]=4 ⇒ cost = 0 ⇒ F[{4}]=0

Kết luận: mọi mask 1 phần tử có F=0.

5) Sinh các trạng thái 2 phần tử (từ từng mask 1 phần tử)

Từ x={1} (0001, S=1, F=0)

- Thêm p=2 → y={1,2} (0011, S=3, k=2)
  - tb=1, r=1
  - M[2]=2, tb < M[2], r>0 ⇒ cost = 2-(1+1)=0
  - F[{1,2}] ≤ 0 + 0 = 0
- Thêm p=3 → y={1,3} (0101, S=4, k=2)
  - tb=2, r=0
  - M[3]=3, tb < M[3], r=0 ⇒ cost = 3-2=1
  - F[{1,3}] ≤ 1
- Thêm p=4 → y={1,4} (1001, S=5, k=2)

- tb=2, r=1
- M[4]=4, tb < M[4], r>0 ⇒ cost = 4-(2+1)=1
- F[{1,4}] ≤ 1

Từ x={2} (0010, S=2, F=0)

- Thêm p=1 → y={1,2} (0011, S=3, k=2)
  - tb=1, r=1
  - M[1]=1, tb >= M[1] ⇒ cost = 1-1=0
  - F[{1,2}] ≤ min(đang 0, 0) = 0
- Thêm p=3 → y={2,3} (0110, S=5, k=2)
  - tb=2, r=1
  - M[3]=3, tb < M[3], r>0 ⇒ cost = 3-(2+1)=0
  - F[{2,3}] ≤ 0
- Thêm p=4 → y={2,4} (1010, S=6, k=2)
  - tb=3, r=0
  - M[4]=4, tb < M[4], r=0 ⇒ cost = 4-3=1
  - F[{2,4}] ≤ 1

Từ x={3} (0100, S=3, F=0)

- Thêm p=1 → y={1,3} (0101, S=4, k=2)
  - tb=2, r=0
  - M[1]=1, tb >= M[1] ⇒ cost = 2-1=1
  - F[{1,3}] ≤ min(đang 1, 1) = 1
- Thêm p=2 → y={2,3} (0110, S=5, k=2)
  - tb=2, r=1
  - M[2]=2, tb >= M[2] ⇒ cost = 2-2=0
  - F[{2,3}] ≤ 0
- Thêm p=4 → y={3,4} (1100, S=7, k=2)
  - tb=3, r=1

    - M[4]=4, tb < M[4], r>0 ⇒ cost = 4-(3+1)=0
    - F[{3,4}] ≤ 0

Từ x={4} (1000, S=4, F=0)

- Thêm p=1 → y={1,4} (1001, S=5, k=2)
    - tb=2, r=1
    - M[1]=1, tb >= M[1] ⇒ cost = 2-1=1
    - F[{1,4}] ≤ 1 (không đổi)
- Thêm p=2 → y={2,4} (1010, S=6, k=2)
    - tb=3, r=0
    - M[2]=2, tb >= M[2] ⇒ cost = 3-2=1
    - F[{2,4}] ≤ 1 (không đổi)
- Thêm p=3 → y={3,4} (1100, S=7, k=2)
    - tb=3, r=1
    - M[3]=3, tb >= M[3] ⇒ cost = 3-3=0
    - F[{3,4}] ≤ 0 (đã 0)

Tóm tắt mask 2 phần tử (giá trị tốt nhất thu được):

- F[{1,2}] = 0, F[{2,3}] = 0, F[{3,4}] = 0
- F[{1,3}] = 1, F[{1,4}] = 1, F[{2,4}] = 1

---

6) Sinh các trạng thái 3 phần tử (từ mask 2 phần tử)

Mình xét những nhánh “rẻ” (F=0) trước: {1,2}, {2,3}, {3,4}.

Từ x={1,2} (0011, S=3, F=0)

- Thêm p=3 → y={1,2,3} (0111, S=6, k=3)
    - tb=2, r=0
    - M[3]=3 ⇒ tb < M[3], r=0 ⇒ cost = 3-2=1
    - F[{1,2,3}] ≤ 1
- Thêm p=4 → y={1,2,4} (1011, S=7, k=3)

- tb=2, r=1
- M[4]=4 ⇒ tb < M[4], r>0 ⇒ cost = 4-(2+1)=1
- F[{1,2,4}] ≤ 1

Từ x={2,3} (0110, S=5, F=0)

- Thêm p=1 → y={1,2,3} (0111, S=6, k=3)
  - tb=2, r=0
  - M[1]=1 ⇒ tb >= M[1] ⇒ cost = 2-1=1
  - F[{1,2,3}] ≤ min(đang 1, 1) = 1
- Thêm p=4 → y={2,3,4} (1110, S=9, k=3)
  - tb=3, r=0
  - M[4]=4 ⇒ tb < M[4], r=0 ⇒ cost = 4-3=1
  - F[{2,3,4}] ≤ 1

Từ x={3,4} (1100, S=7, F=0)

- Thêm p=1 → y={1,3,4} (1101, S=8, k=3)
  - tb=2, r=2
  - M[1]=1 ⇒ tb >= M[1] ⇒ cost = 2-1=1
  - F[{1,3,4}] ≤ 1
- Thêm p=2 → y={2,3,4} (1110, S=9, k=3)
  - tb=3, r=0
  - M[2]=2 ⇒ tb >= M[2] ⇒ cost = 3-2=1
  - F[{2,3,4}] ≤ min(đang 1, 1)=1

Tóm tắt mask 3 phần tử (giá trị tốt nhất):

- F[{1,2,3}] = 1
- F[{1,2,4}] = 1
- F[{1,3,4}] = 1
- F[{2,3,4}] = 1

---

7) Trạng thái đủ 4 bao (full = 1111)

Giờ từ các mask 3 phần tử, thêm bao còn thiếu để ra full:

Nhánh tốt 1: {1,2} (F=0) → {1,2,4} (F=1) → thêm p=3 → full

- Từ {1,2,4} (S=7, k=3, F=1), thêm p=3:
  - y=1111, S=10, k=4
  - tb=10/4=2, r=2
  - M[3]=3, tb < M[3] và r>0 ⇒ cost = 3-(2+1)=0
  - Tổng chi phí = 1 + 0 = 1

Nhánh tốt 2: {1}→{1,2}→{1,2,4}→{1,2,3,4} giống trên, cũng 1.

Nhánh kém hơn để đối chứng: {2,3} (F=0) → {1,2,3} (F=1) → thêm p=4

- Từ {1,2,3} (S=6, k=3, F=1), thêm p=4:
  - y=1111, S=10, k=4
  - tb=2, r=2
  - M[4]=4, tb < M[4], r>0 ⇒ cost = 4-(2+1)=1
  - Tổng chi phí = 1 + 1 = 2 (kém)

Kết quả DP lấy min qua tất cả con đường:
F[1111] = 1 ⇒ in ra **1** (khớp đề bài).