

NAMUNAVIY MASALALAR

1-masala. Ikki element yig'indisi

Massiv va S soni berilgan. Shunday ikki element bormi-yo'qmi, ularning yig'indisi S ga teng bo'lsin?

Yechim (two pointers, $O(n \log n)$)

```
n, S = map(int, input().split())
a = list(map(int, input().split()))

a.sort()
l, r = 0, n - 1
ok = False

while l < r:
    s = a[l] + a[r]
    if s == S:
        ok = True
        break
    if s < S:
        l += 1
    else:
        r -= 1

print("YES" if ok else "NO")
```

2-masala. Oraliq yig'indilar

Massiv va Q ta so'rov berilgan. Har so'rovda $l \ r$ — l dan r gacha yig'indini toping.

Yechim (prefix sum, $O(n + Q)$)

```
n = int(input())
a = list(map(int, input().split()))

pref = [0] * (n + 1)
for i in range(n):
    pref[i + 1] = pref[i] + a[i]

q = int(input())
for _ in range(q):
    l, r = map(int, input().split())
    print(pref[r] - pref[l - 1])
```

3-masala. Minimal bo'lak ($\geq S$)

Musbat sonlardan iborat massiv va S berilgan.

Yig'indisi kamida S bo'lgan eng kichik uzluksiz bo'lak uzunligini toping.

Agar yo'q bo'lsa 0 .

Yechim (sliding window, $O(n)$)

```
n, S = map(int, input().split())
a = list(map(int, input().split()))

l = 0
cur = 0
ans = n + 1

for r in range(n):
    cur += a[r]
    while cur >= S:
        ans = min(ans, r - l + 1)
        cur -= a[l]
        l += 1

print(ans if ans <= n else 0)
```

4-masala. Eng uzun noyob bo‘lak

Massiv berilgan.

Takrorlanmagan elementlardan iborat eng uzun uzluksiz bo‘lak uzunligini toping.

Yechim (set + two pointers)

```
n = int(input())
a = list(map(int, input().split()))

seen = set()
l = 0
ans = 0

for r in range(n):
    while a[r] in seen:
        seen.remove(a[l])
        l += 1
    seen.add(a[r])
    ans = max(ans, r - l + 1)

print(ans)
```

5-masala. Binary search — eng yaqin element

Saralangan massiv va X berilgan. X ga eng yaqin elementni toping (teng masofada bo‘lsa — kichigi).

Yechim (bisect, $O(\log n)$)

```
import bisect

n = int(input())
a = list(map(int, input().split()))
```

```

x = int(input())

pos = bisect.bisect_left(a, x)

if pos == 0:
    print(a[0])
elif pos == n:
    print(a[-1])
else:
    if abs(a[pos] - x) < abs(a[pos - 1] - x):
        print(a[pos])
    else:
        print(a[pos - 1])

```

6-masala. Greedy — vazifalar soni

n ta vazifa va umumiy vaqt T berilgan. Har vazifaning vaqti ma'lum. Ketma-ket bajarib, eng ko'p vazifani bajaring.

Yechim (greedy + sort)

```

n, T = map(int, input().split())
t = list(map(int, input().split()))

t.sort()
cur = 0
cnt = 0
for x in t:
    if cur + x <= T:
        cur += x
        cnt += 1
    else:
        break

print(cnt)

```

7-masala. BFS — eng qisqa masofa

Graf (n tugun, m qirra) berilgan.

1-tugundan n -tugungacha eng qisqa masofani toping (og'irliksiz).

Yechim (BFS)

```

from collections import deque

n, m = map(int, input().split())
g = [[] for _ in range(n + 1)]
for _ in range(m):
    u, v = map(int, input().split())
    g[u].append(v)
    g[v].append(u)

dist = [-1] * (n + 1)

```

```

dq = deque([1])
dist[1] = 0

while dq:
    u = dq.popleft()
    for v in g[u]:
        if dist[v] == -1:
            dist[v] = dist[u] + 1
            dq.append(v)

print(dist[n])

```

8-masala. DFS — komponentlar soni

Graf berilgan. Bog‘langan komponentlar sonini toping.

Yechim (DFS)

```

import sys
sys.setrecursionlimit(10**7)

n, m = map(int, input().split())
g = [[] for _ in range(n + 1)]
for _ in range(m):
    u, v = map(int, input().split())
    g[u].append(v)
    g[v].append(u)

vis = [False] * (n + 1)

def dfs(u):
    vis[u] = True
    for v in g[u]:
        if not vis[v]:
            dfs(v)

cnt = 0
for i in range(1, n + 1):
    if not vis[i]:
        dfs(i)
        cnt += 1

print(cnt)

```

9-masala. DP — zinadan chiqish

n pog‘onali zina bor.
 1 yoki 2 qadam tashlab chiqish mumkin.
 Nechta usul bor?

Yechim (DP, $O(n)$)

```

n = int(input())

if n == 0:
    print(1)
else:
    dp = [0] * (n + 1)
    dp[0] = 1
    dp[1] = 1
    for i in range(2, n + 1):
        dp[i] = dp[i - 1] + dp[i - 2]
    print(dp[n])

```

10-masala. Ko'chma maksimum

Massiv va oynacha o'lchami K berilgan. Har bir oynacha uchun maksimal elementni chiqaring.

Yechim (deque, $O(n)$)

```

from collections import deque

n, k = map(int, input().split())
a = list(map(int, input().split()))

dq = deque()
res = []

for i in range(n):
    while dq and dq[0] <= i - k:
        dq.popleft()
    while dq and a[dq[-1]] <= a[i]:
        dq.pop()
    dq.append(i)
    if i >= k - 1:
        res.append(a[dq[0]])

print(*res)

```