Resolução de Problemas

merge sort tree e programação dinâmica

Exercício A - Frog Jumping

Resolução em Python:

```python
t = int(input())
for _ in range(t):
    a, b, k = map(int, input().split())
    if k % 2 == 0:
        print((a - b) * (k // 2))
    else:
        print((a - b) * (k // 2) + a)
```

Exercício B - Vacations

Resolução em Python:

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

dp = [[float('inf')] * 3 for _ in range(n)]

if a[0] == 0:
    dp[0][0] = 1
elif a[0] == 1:
    dp[0][0] = 1
    dp[0][1] = 0
elif a[0] == 2:
    dp[0][0] = 1
    dp[0][2] = 0
elif a[0] == 3:
    dp[0][0] = 1
    dp[0][1] = 0
    dp[0][2] = 0

for i in range(1, n):
    current = a[i]
    dp[i][0] = 1 + min(dp[i-1][0], dp[i-1][1], dp[i-1][2])

    if current == 1 or current == 3:
        dp[i][1] = min(dp[i-1][0], dp[i-1][2])

    if current == 2 or current == 3:
        dp[i][2] = min(dp[i-1][0], dp[i-1][1])

result = min(dp[n-1][0], dp[n-1][1], dp[n-1][2])
print(result)
```

Exercicio C - Not So Simple Polygon Embedding

Resolução em Python:

```python
import math

T = int(input())
for _ in range(T):
    n = int(input())
    angle = math.pi / (4 * n)
    radius = 1 / (2 * math.sin(math.pi / (2 * n)))
    side_length = 2 * radius * math.cos(angle)
    print("{0:.9f}".format(side_length))
```

Exercício D - Enemy is Week

Resolução em Python:

```python
import sys

def main():
    input = sys.stdin.readline
    n = int(input())
    a = list(map(int, input().split()))
    vals = sorted(a)
    rnk = [0] * n
    rank = {v: i+1 for i, v in enumerate(vals)}
    for i, v in enumerate(a):
        rnk[i] = rank[v]

    fw1 = [0] * (n + 1)
    leftGreater = [0] * n
    for j in range(n):
        pos = rnk[j]
        s = 0
        i = pos
        while i > 0:
            s += fw1[i]
            i -= i & -i
        leftGreater[j] = j - s
        i = pos
        while i <= n:
            fw1[i] += 1
            i += i & -i
    fw2 = [0] * (n + 1)
    ans = 0
    for j in range(n - 1, -1, -1):
        pos = rnk[j]
        s = 0
        i = pos - 1
        while i > 0:
            s += fw2[i]
            i -= i & -i
        ans += leftGreater[j] * s
        i = pos
        while i <= n:
            fw2[i] += 1
            i += i & -i

    print(ans)

if __name__ == '__main__':
    main()
```

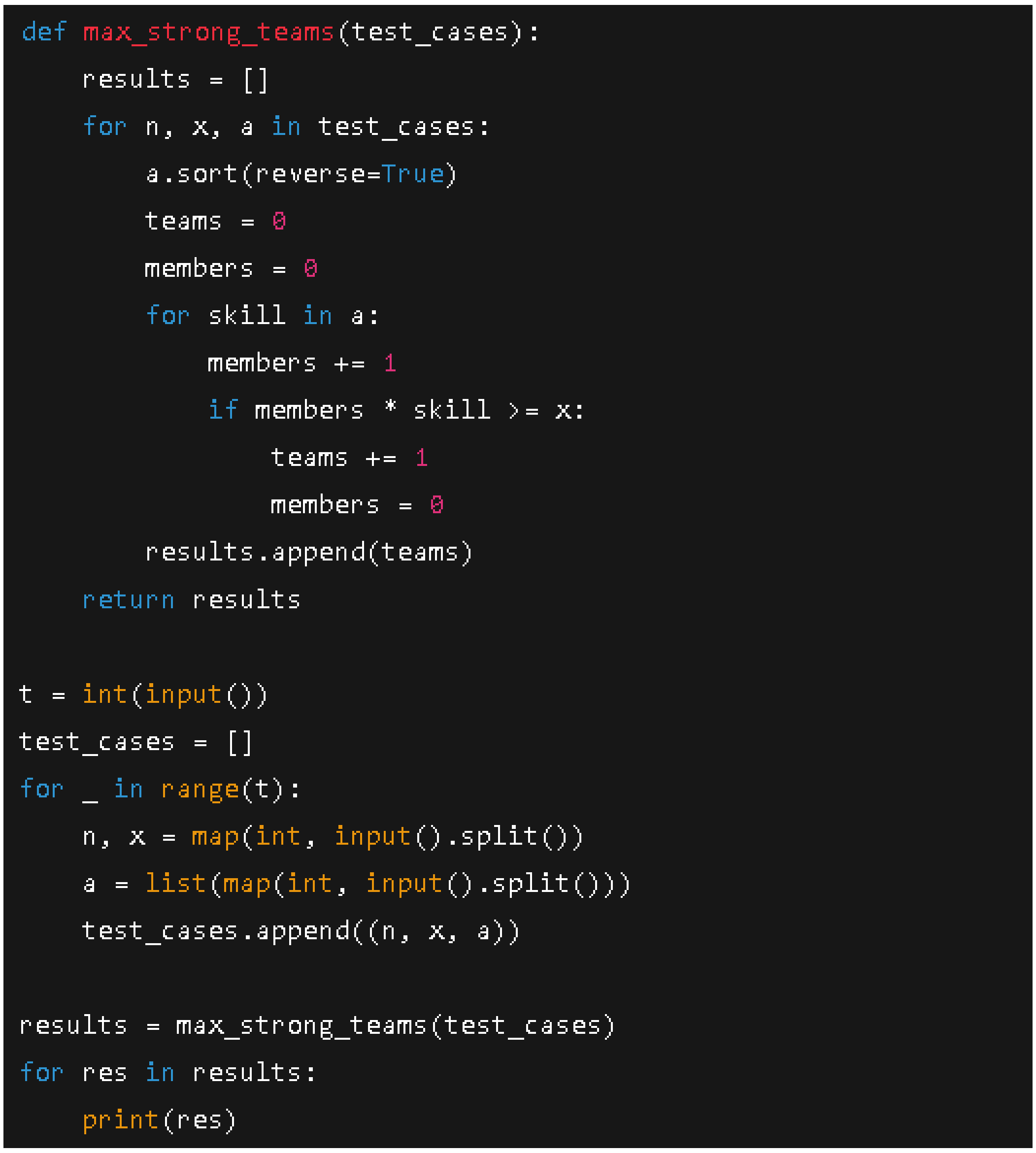
Exercício E - Vanya and Exams

Resolução em Python:

```python
import sys

def main():
    input = sys.stdin.readline
    n, r, avg = map(int, input().split())
    exams = []
    total = 0
    for _ in range(n):
        a, b = map(int, input().split())
        exams.append((b, a))
        total += a

    need = avg * n - total
    if need <= 0:
        print(0)
        return

    exams.sort()
    essays = 0
    for cost, grade in exams:
        add = min(r - grade, need)
        essays += add * cost
        need -= add
        if need == 0:
            break

    print(essays)

if __name__ == '__main__':
    main()
```

Exercício F - Team Training

Resolução em Python:



```python
def max_strong_teams(test_cases):
    results = []
    for n, x, a in test_cases:
        a.sort(reverse=True)
        teams = 0
        members = 0
        for skill in a:
            members += 1
            if members * skill >= x:
                teams += 1
                members = 0
        results.append(teams)
    return results

t = int(input())
test_cases = []
for _ in range(t):
    n, x = map(int, input().split())
    a = list(map(int, input().split()))
    test_cases.append((n, x, a))

results = max_strong_teams(test_cases)
for res in results:
    print(res)
```