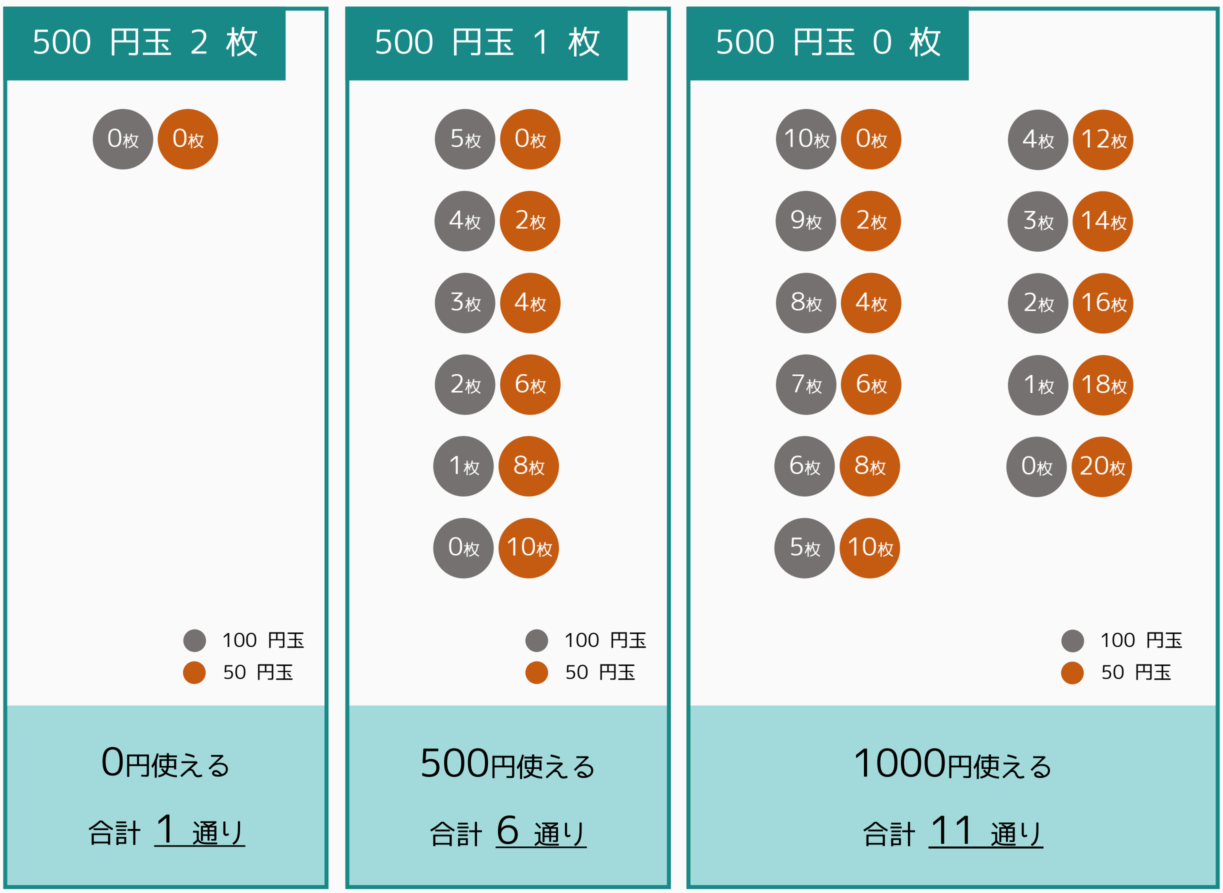
5.6 節末問題 5.6 の解答

問題 5.6.1 (1), (2), (3)

500 円玉を 2 枚・1 枚・0 枚使ったとき、100 円玉と 50 円玉合わせた金額はそれぞれ 0 円・500 円・1000 円となります。

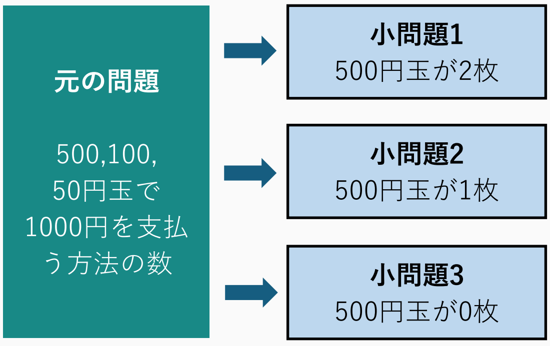
したがって、それぞれの支払う方法の数は下図に示すとおりです。



問題 5.6.1 (4)

問題を右図のように「500 円玉の枚数」で分解することを考えます。

そうすると、(1), (2), (3) の結果より、答えが $1+6+11=$ **18 通り**であることが分かります。



問題 5.6.2

まず、問題を以下のように分解することを考えましょう。

- 小問題 1：選んだ整数の最大値が A_1 となる選び方は何通り？
- 小問題 2：選んだ整数の最大値が A_2 となる選び方は何通り？
- 小問題 3：選んだ整数の最大値が A_3 となる選び方は何通り？
- ：
- 小問題 N：選んだ整数の最大値が A_N となる選び方は何通り？

このとき、求めるべき答えは以下のようになります。

> （小問題 1 の答え）× $A_1 + \cdots +$（小問題 N の答え）× A_N

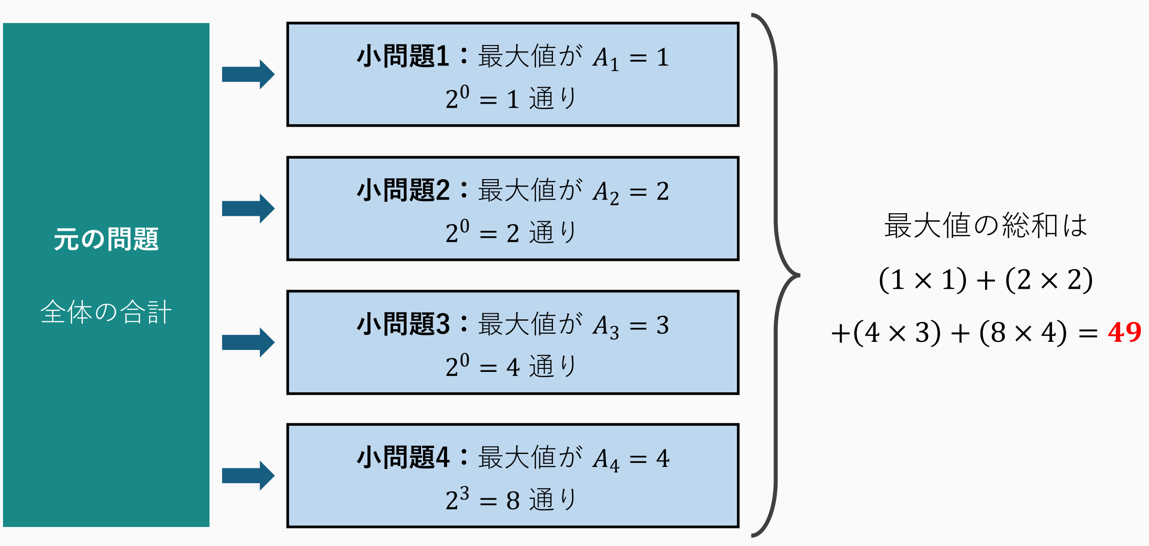
そこで、選んだ整数の最大値が A_i となるような選び方の条件は以下のとおりです。

- A_i を選ぶ。
- $A_1, A_2, A_3, \ldots, A_{i-1}$ の中から 0 個以上を選ぶ。

$i-1$ 個のものについて Yes/No を選択できるので、積の法則（→**3.3.2項**）より選び方は全部で 2^{i-1} 通りあります。したがって、求める答えは以下の通りです。

$$\sum_{i=1}^{N} 2^{i-1} \times A_i = (2^0 \times A_1) + (2^1 \times A_2) + \cdots + (2^{N-1} \times A_N)$$

たとえば、$N = 4, (A_1, A_2, A_3, A_4) = (1, 2, 3, 4)$ の場合、下図のようにして答えが 49 だと分かります。



これをプログラムで実装すると、以下のようになります。なお、変数 `power[i]` は 2^i を 1000000007 で割った余りとなっています。

```cpp
#include <iostream>
using namespace std;

const long long mod = 1000000007;
long long N;
long long A[300009];
long long power[300009];

int main() {
    // 入力
    cin >> N;
    for (int i = 1; i <= N; i++) cin >> A[i];

    // 2^i を求める
    power[0] = 1;
    for (int i = 1; i <= N; i++) {
        power[i] = (2 * power[i - 1]) % mod;
    }

    // 答えを求める
    long long Answer = 0;
    for (int i = 1; i <= N; i++) {
        Answer += power[i - 1] * A[i];
        Answer %= mod;
    }

    // 出力
    cout << Answer << endl;
    return 0;
}
```

※ Python などのソースコードは chap5-6.md をご覧ください。