# Алгоритм бінарного піднесення в степінь

Підготував Грибок Андрій

5 курс ФВЄ

# Швидке піднесення в степінь

- **Повторюване піднесення до квадрата** — алгоритм, призначений для піднесення числа $x$ до натурального степеня $n$ за менше число множень, ніж цього вимагає визначення степені.
- Алгоритм не завжди найоптимальніший: наприклад, піднесення в степінь $n = 15$ повторюваним піднесенням до квадрата потребує 6 множень, хоча це можна досягти за 5.

- Піднесення до степеня $a^n$ відбувається за рахунок множення a на n-1 таких же a.
- Але це можна зробити набагато швидшим шляхом:
- Ми знаємо, що $a^{b+c}=a^b \cdot a^c$ і $a^{2b}=a^b \cdot a^b$
- Найкращий спосіб знайти ці множення це представити степінь в двійковій системі

# Наприклад

$3^{11}=3^{1011_2}=3^8\cdot 3^2\cdot 3^1$

Отже, нам потрібно лише знайти послідовність чисел, в якій кожен наступний член квадрат попереднього:

$3^1=3$

$3^2=(3^1)^2=3\cdot 3=9$

$3^4=(3^2)^2=9\cdot 9=81$

$3^8=(3^4)^2=81\cdot 81=6561$

Ми отримуємо кінцевий результат перемножуючи три з них, тому що біля $3^4$ стоїть 0.

$3^{11}=6561\cdot 9\cdot 3=177147$.

# Реалізація алгоритму

- Використовуючи нашу асоціативність ми знаємо що робити якщо степінь кратний двом

$a^{2b}=a^b \cdot a^b$

- Якщо ж ні то просто виділяємо один степінь :

$a^n=a^{n-1} \cdot a$

- Ітак ми по суті уже маємо рекурентну формулу в якій буде не більше ніж 2 log n множень

# Реалізація (рекурсивний метод)

```
int binpow (int a, int n) {
        if (n == 0)
                return 1;
        if (n % 2 == 1)
                return binpow (a, n-1) * a;
        else {
                int b = binpow (a, n/2);
                return b * b;
        }
}
```

# Заміна деяких обчислень бітовими операціями

```
int binpow (int a, int n) {
        int res = 1;
        while (n)
                if (n & 1) {
                        res *= a;
                        --n;
                }
                else {
                        a *= a;
                        n >>= 1;
                }
        return res;
}
```

```
int binpow (int a, int n)
{ int res = 1;
     while (n)
          { if (n & 1)
          res ·= a;
          a ·= a;
          n >>= 1; }
return res;
```

# Використання

- Ефективне обчислення чисел Фібоначчі
- Швидке застосування набору геометричних операцій до точок
- Кількість шляхів фіксованої довжини в графі
- Варіація бінарного зведення в ступінь: множення двох чисел по модулю