Алгоритмы и структуры данных-1

SET 1. Задача A2b.

Осень 2024. Клычков М. Д.

```
int fastExponent(int x, int n) {
  int r = 1;
  int p = x;
  int e = n;

  while (e > 0) {
    if (e % 2 != 0) {
      r *= p;
    }
    p *= p;
    e /= 2;
  }

  return r;
}
```

Пункт 1. Этот алгоритм очень похож на перевод десятичного числа n в двоичную систему счисления. Буквально, мы перебираем с помощью переменной p все степени двойки (здесь они представлены как $\{x^1, x^2, x^4, x^8, x^{16}, \dots\}$) и, если этот разряд необходим для представления числа (то есть остаток от деления — 1), то добавляем к результату. Другими словами, $r = x^{2^{i_1}+2^{i_2}+\dots+2^{i_k}}$. Тогда в теле `if` произойдет количество умножений равное количеству единиц в двоичной записи числа n.

Цель: найти число, описывающее количество единиц в двоичной записи числа n — $f(n)$. Пусть функция $r(n)$ — остаток n при делении на 2. Тогда:

$$f(n) = r(n) + r\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + r\left(\left\lfloor \frac{n}{4} \right\rfloor\right) + r\left(\left\lfloor \frac{n}{8} \right\rfloor\right) + \dots$$

Это полностью соответствует алгоритму поиска представления числа в двоичной системе счисления «в столбик».

$$r(n) = n\%2 = n - 2\left\lfloor \frac{n}{2} \right\rfloor$$

Подставим и получим:

$$\begin{aligned} f(n) &= n - 2\left\lfloor \frac{n}{2} \right\rfloor + \left\lfloor \frac{n}{2} \right\rfloor - 2\left\lfloor \frac{n}{4} \right\rfloor + \left\lfloor \frac{n}{4} \right\rfloor - 2\left\lfloor \frac{n}{8} \right\rfloor + \dots = \\ &= n - \sum_{i=1}^{\lfloor \log_2 n \rfloor} \left\lfloor \frac{n}{2^i} \right\rfloor \end{aligned}$$

Учитывая еще, что всего в тело цикла будет $(\lfloor \log_2 n \rfloor + 1)$ вхождений, можно наконец составить общее количество умножений:

$$\begin{aligned} m(n) &= (\lfloor \log_2 n \rfloor + 1) + f(n) = \\ &= (\lfloor \log_2 n \rfloor + 1) + n - \sum_{i=1}^{\lfloor \log_2 n \rfloor} \left\lfloor \frac{n}{2^i} \right\rfloor = \\ &= n + \lfloor \log_2 n \rfloor - \sum_{i=1}^{\lfloor \log_2 n \rfloor} \left\lfloor \frac{n}{2^i} \right\rfloor + 1 \end{aligned}$$

В задаче просили именно точное количество умножений, поэтому как-то усреднять это значения и приводить ответ к другому виду не имеет смысла.

Пункт 2. Выберем в качестве инварианта цикла `while` условие:

$$r \cdot p^e = x^n$$

Такой выбор мотивирован той же идеей, что и в пункте 1, а именно переводом числа n в двоичную систему счисления. На i-й итерации цикла (отсчет с 0) мы принимаем решение об i-м бите числа n. Если в двоичной записи n должен присутствовать i-й бит, то изменяется результирующая переменная r, которая накапливает уже собранные степени двойки.

Можно по-другому представить, что происходит, например, так:

$$\begin{aligned}
r \cdot p^e &= x^n \\
1 \cdot x^n &= x^n \\
(1 \cdot x^{n\%2}) \cdot \left(x^2\right)^{\left\lfloor \frac{n}{2} \right\rfloor} &= x^n \\
(1 \cdot x^{n\%2} \cdot x^{\left\lfloor \frac{n}{2} \right\rfloor \%2}) \cdot \left(x^4\right)^{\left\lfloor \frac{\left\lfloor \frac{n}{2} \right\rfloor}{2} \right\rfloor} &= x^n \\
\ldots \cdot \ldots &= \ldots \\
x^n \cdot \left(x^{2^{\lfloor \log_2 n \rfloor}}\right)^0 &= x^n
\end{aligned}$$

`INIT`: Перед циклом переменные проинициализировались так: $r \coloneqq 1$; $p \coloneqq x$; $e \coloneqq n$. Осталось лишь подставить: $r \cdot p^e = 1 \cdot x^n = x^n$.

`MNT`: Пусть на i-ой итерации цикла: $x^k \cdot \left(x^{2^i}\right)^m = x^n$, тогда:

$$\begin{aligned}
x^k \cdot \left(x^{2^i}\right)^m &= x^n \\
x^k \cdot x^{m \cdot 2^i} &= x^n \\
x^k \cdot x^{\left(2\left\lfloor \frac{m}{2} \right\rfloor + m\%2\right) \cdot 2^i} &= x^n \\
x^k \cdot x^{\left(2\left\lfloor \frac{m}{2} \right\rfloor + m\%2\right) \cdot 2^i} &= x^n \\
x^k \cdot x^{2^{i+1}\left\lfloor \frac{m}{2} \right\rfloor + 2^i(m\%2)} &= x^n \\
x^{k+2^i(m\%2)} \cdot x^{2^{i+1}\left\lfloor \frac{m}{2} \right\rfloor} &= x^n
\end{aligned}$$

Преобразовав исходное выражение нашли новые m и k, следовательно на промежуточных шагах инвариант поддерживается.

`TRM`: Причина выхода из цикла: $e = 0$. Тогда:

$$r \cdot p^0 = x^n \Rightarrow r = x^n$$