

Модели

RAM-модель (Random Access Machine) Вопросы, возникающие при создании модели

1. адресация
2. какие инструкции
3. рекурсия
4. где лежат инструкции
5. размер данных
6. кол-во памяти
7. случайность

Адресация Есть ячейки, в которых можно хранить целые числа (ограничения на $MAXC$ разумные, и на них введена неявная адресация

Замечание. Явная адресация — при создании элемента получаем адрес и можем пользоваться только этим адресом. Неявно — можем получать адреса каким-то своим образом, к примеру, $ptr + 20$.

Кол-во памяти Неявное соглашение RAM — время работы не меньше памяти. По дефолту считаем, что мы его инициализируем мусором

Где инструкции Хранить инструкции можно в памяти и где-то снаружи. Мы будем хранить снаружи (внутри — RASP-модель). Иначе говоря, инструкции и данные отделены.

Какие инструкции В нашей модели есть инструкции следующих типов:

- работа с памятью
- ветвление
- передача управления ($=goto$),
- арифметика (at least $a + b, a - b, \frac{a}{b}, \cdot, mod, \lfloor \frac{a}{b} \rfloor$)
- сравнения (at least $a < b, a > b, a \leq b, a \geq b, a = b, a \neq b$)
- логические (at least $\wedge, \vee, \oplus, \neg$)
- битовые операции ($>>, <<, \&, |, \sim, \oplus$)
- математические функции (опять-таки, в рамках разумного)
- rand

Все инструкции работают от конечного разумного числа операндов (не умеем в векторные операции)

Размер данных $\exists C, k : C \cdot A^k \cdot n^k$ — верхнее ограничение на величины промежуточных вычислений.

Рекурсия Рекурсия всегда линейна по памяти относительно глубины.

Случайность Мы считаем, что у нас есть абсолютно случайная функция. Будем полагать, что у нас есть источник энтропии, выдающий случайности в промежутке $[0, 1]$.

Время работы.

- наихудшее — $t = \max_{input, random} t(input, random)$
- наилучшее — $t = \max_{input} \min_{random} t(input, random)$
- ожидаемое — $E t = \max_{input} Average_{random} t(input, random)$
- на случайных данных — $t = Average_{input} Average_{random} t(input, random)$

Алгоритмы

Методы доказательства корректности алгоритма.

1. индукция
2. инвариант
3. от противного