CS 入門 4b クラス 第8回資料 (2012.11.22)

担当: 伊知地　宏

1　前回の問題の解答

[問題 7.1] ユークリッドの互除法で最大公約数を求める関数 gcd(x,y) を基に，計算の繰り返し数を返す関数 gcd_bench(x,y) を作りなさい．

```
def gcd_bench(x,y)
  r = x % y
  times = 1
  while r > 0
    x = y
    y = r
    r = x % y
    times = times + 1
  end
  times
end
```

[問題 7.2] どのような場合に，ユークリッドの互除法でも計算回数が増えるのか考察しなさい．

gcd2_bench(144, 89) の値は 10 となる．その周辺では，gcd2_bench(146, 89)，gcd2_bench(153, 89) の値が 8 になる．(146, 89) と (153, 89) という組の場合にも，ユークリッドの互除法で計算を行うと，計算の途中経過で割り算の商のほとんどが 1 となる．

フィボナッチ数の隣り合う 2 数の最大公約数を求める場合に，ユークリッドの互除法の計算が最も長くなるが，それだけでなく，最大公約数が 1 でユークリッドの互除法での計算で商が 1 の場合が多い数の組でも計算の手間がかかる．

[問題 7.3] フィボナッチ数の計算を行う計算量 O(n) の関数 fib(n) を作りなさい．

```
def fib(n)
  f = 1
  p1 = 1
  for k in 2..n
    p2 = p1
    p1 = f
    f = p1 + p2
  end
  f
end
```

[問題 7.4]

1. **n の k 乗を n を単純に k 回掛けて計算する関数 power1(n, k) を作りなさい．**
2. **n の k 乗をもっと効率よく計算する関数 power2(n, k) を作りなさい．**
3. **関数 power1(n, k) を基に，計算の繰り返し回数を返す関数 power1_bench(n, k) を作りなさい．**
4. **関数 power2(n, k) を基に，計算の繰り返し回数を返す関数 power2_bench(n ,k) を作りなさい．**

```
def power1(n, k)
  ans = 1
  for i in 1..k
    ans = n * ans
  end
  ans
end

def power2(n, k)
  ans = 1
  while k > 0
    times = 1
    sub_ans = n
    while k >= 2 * times
      sub_ans = sub_ans * sub_ans
```

```
      times = 2 * times
    end
    ans = ans * sub_ans
    k = k - times
  end
  ans
end

def power1_bench(n, k)
  count = 0
  ans = 1
  for i in 1..k
    ans = n * ans
    count = count + 1
  end
  count
end

def power2_bench(n, k)
  count = 0
  ans = 1
  while k > 0
    times = 1
    sub_ans = n
    while k >= 2 * times
      sub_ans = sub_ans * sub_ans
      times = 2 * times
      count = count + 1
    end
    ans = ans * sub_ans
    k = k - times
    count = count + 1
  end
  count
end
```

[問題 7.5] 関数 power1_bench(n, k)，関数 power2_bench(n ,k) で実験を行い，関数 power1(n, k) のアルゴリズムの計算量，関数 power2(n, k) のアルゴリズムの計算量を予想しなさい．

power1(n, k) の計算量は明らかに $O(k)$ である．

$C(\text{power2(n, k)})$ で power2(n, k) の計算回数を表すとする．

$k = 2^m$ のときに

$$C(\text{power2(n, k)}) = \log_2 k + 1$$

である．

$k < 2^m$ のときに

$$C(\text{power2(n, k)}) \leq \frac{m(m+1)}{2}$$

となるので，

$$C(\text{power2(n, k)}) \leq \frac{(\log_2 k)^2 + \log_2 k}{2}$$

であり，power2(n, k) の計算量は $O((\log_2 k)^2)$ となる．

2 ソート

ソートとは，データの集まりをある規則で順に並べ替えることである．代表的なソートアルゴリズムとして以下の物がある．

1. 単純ソート
2. マージソート
3. クイックソート

2.1 単純ソート

単純ソートは，全ての要素同士を比較して並べ替えるソートアルゴリズムである．

```
def simplesort(a)
  for i in 0..a.length-2
    for j in i+1..a.length-1
      if a[i] > a[j]
        a[i], a[j] = a[j], a[i]
      end
    end
  end
  a
end
```