プログラミング技法

第2回演習

資料は

http://www-hiraki.is.s.u-tokyo.ac.jp/lectures/prog_giho/

今日の目標

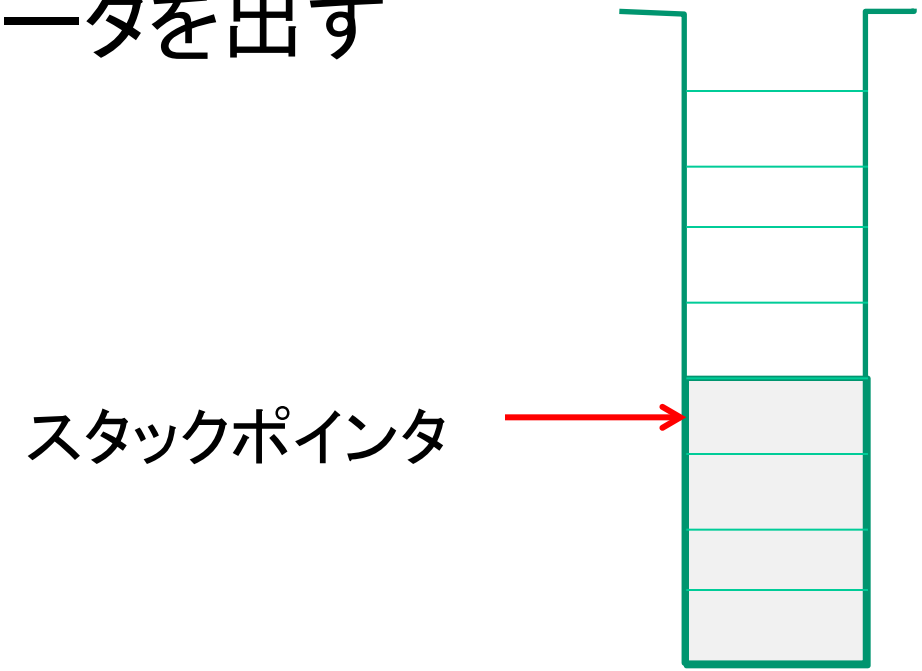
- コンピュータの実行速度の変化を実感する
- 実行時間の測り方になれる
- 言語による振舞いの差を体感する

問題1

- 十分に大きい配列（例えば1000万要素）を取りスタックとして使う。つまり、操作する場所をスタックポインタとしてもつ。
- まず、配列にランダムまたは連続数で順次書き込む（プッシュ操作）。
- 次に配列を一番最近書き込んだものから順次読み出し総和を得る。
- 配列サイズを16から2倍づつ大きくしたときの要素あたりの実行時間変化をリストまたはグラフとして求めよ。

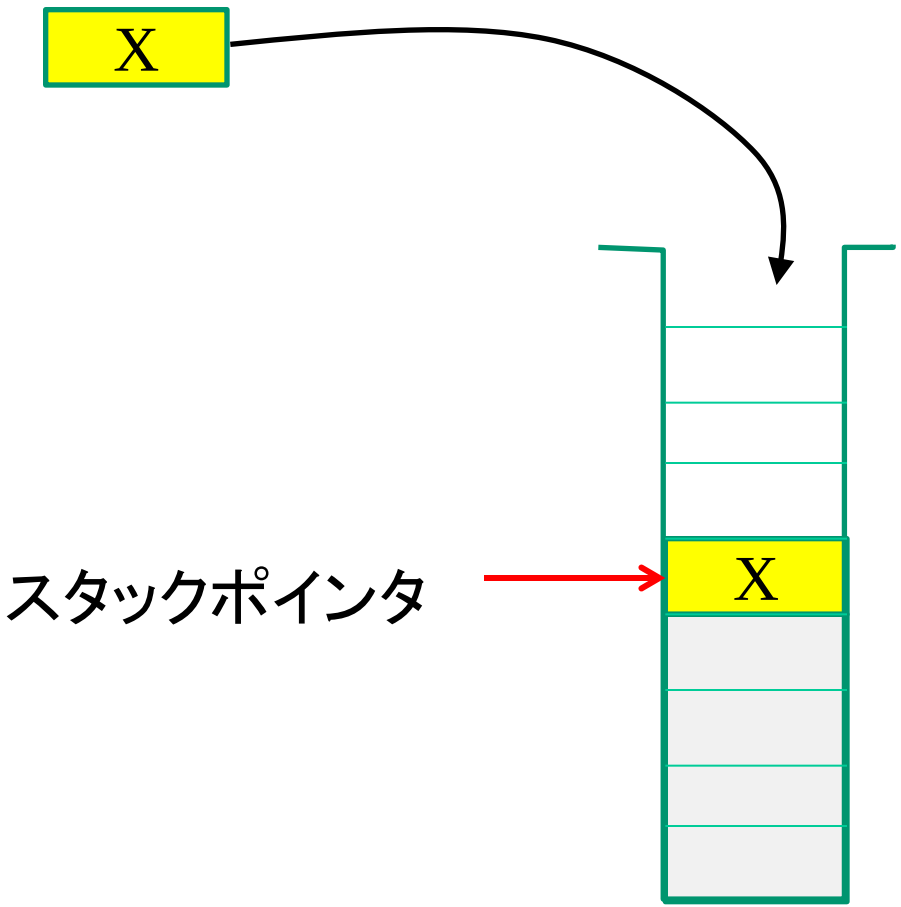
スタック

- メモリを穴のようにつかう。出入り口は一つ
- PUSH操作　　穴にデータを入れる
- POP操作　穴からデータを出す



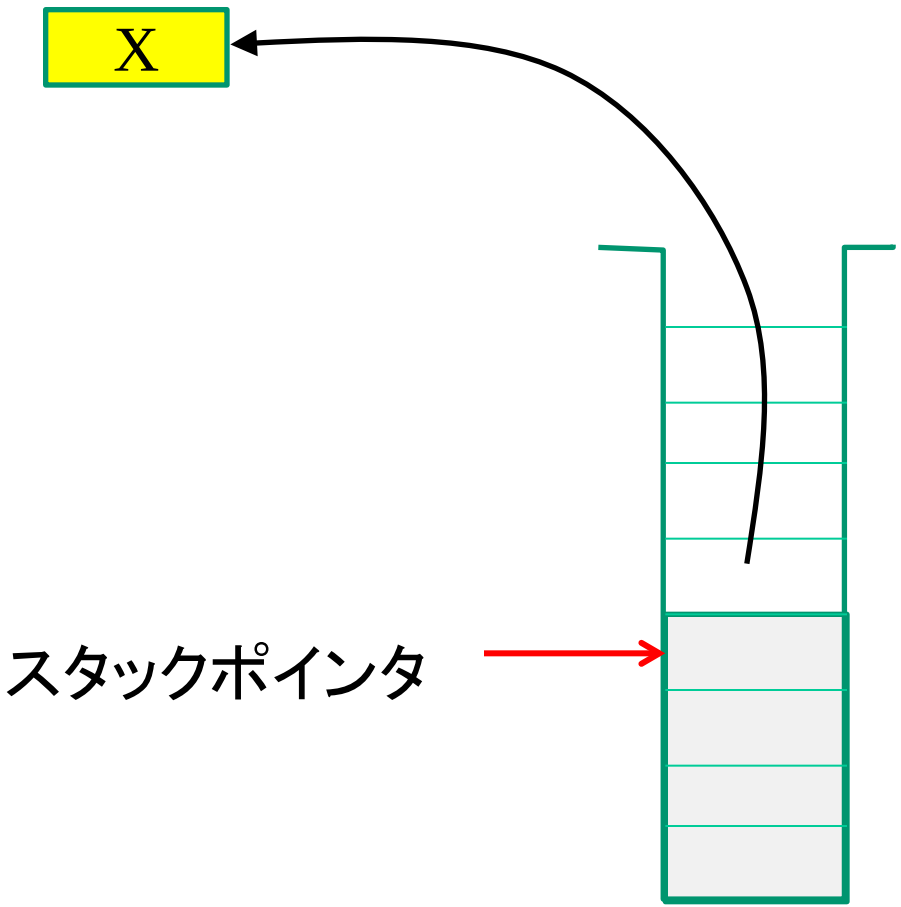
スタック

- PUSH　X



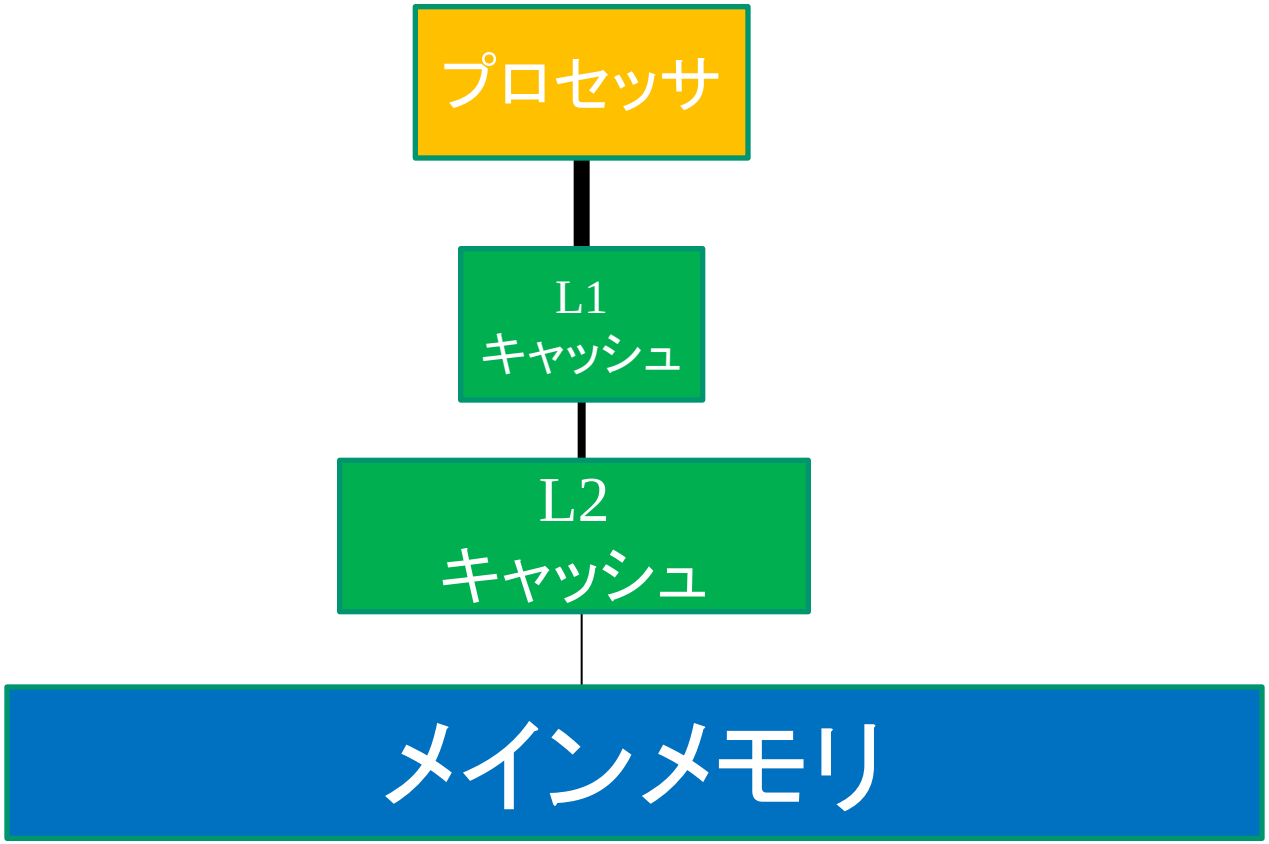
スタック

- POP



何故実行時間が変わるか

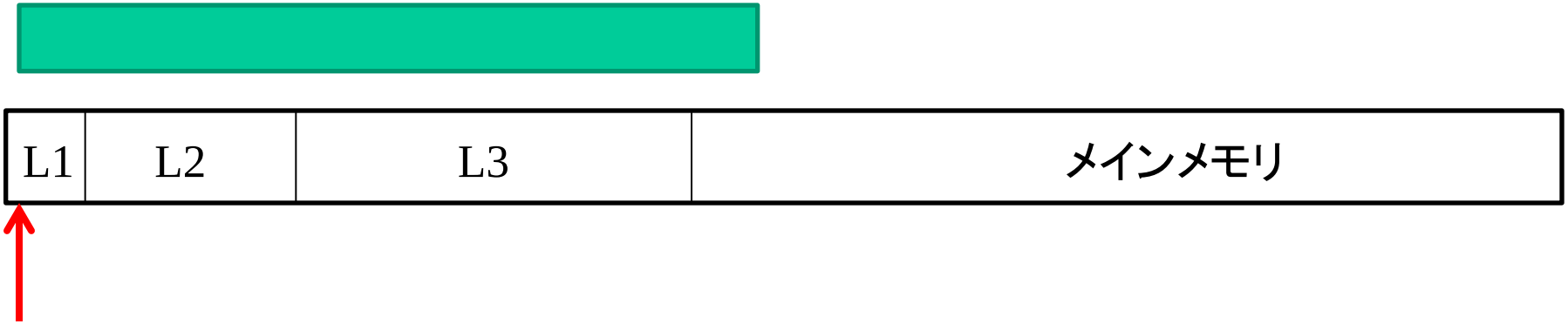
- 階層メモリは下に行くほど低性能



階層メモリへの格納

- スタックが大きくなると上の階層では入らない

スタックの実体：あふれたものが順次下の階層に行く



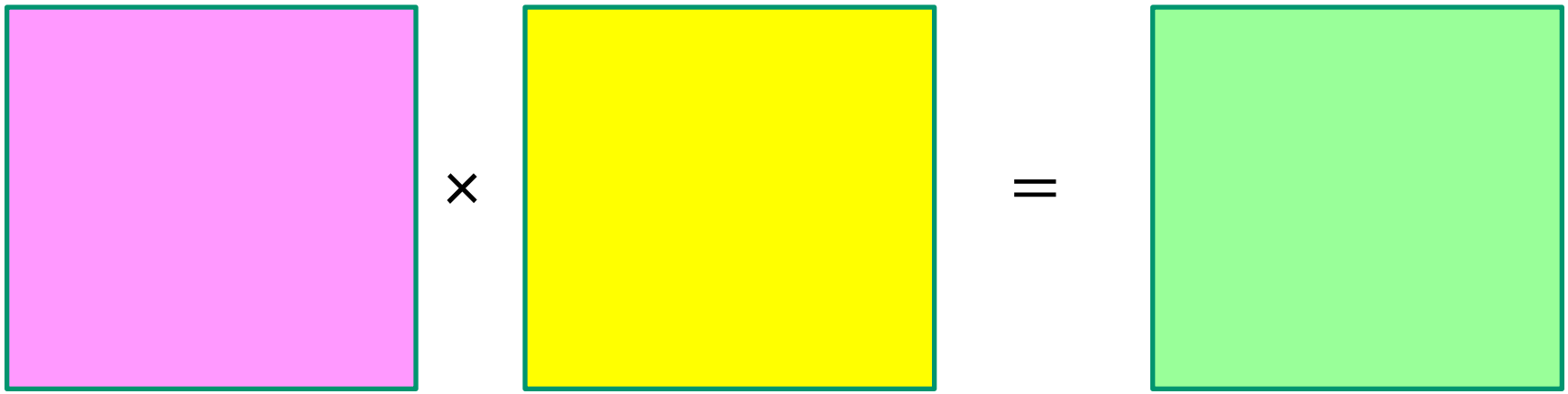
問題2

- 十分に大きい2個の行列の行列積を求め、別の行列として作成するプログラムを作成する。

- 行列サイズを順次大きくし、演算あたりの実行時間変化をリストまたはグラフとして求めよ。

問題1の計算本体部分だけを変更すれば出来る

- これが、どのようにメモリに乗るのか想像しよう
- ループの i,j,kの順を変えたりブロック化して
  振る舞いが変わるか試そう



プログラミング言語

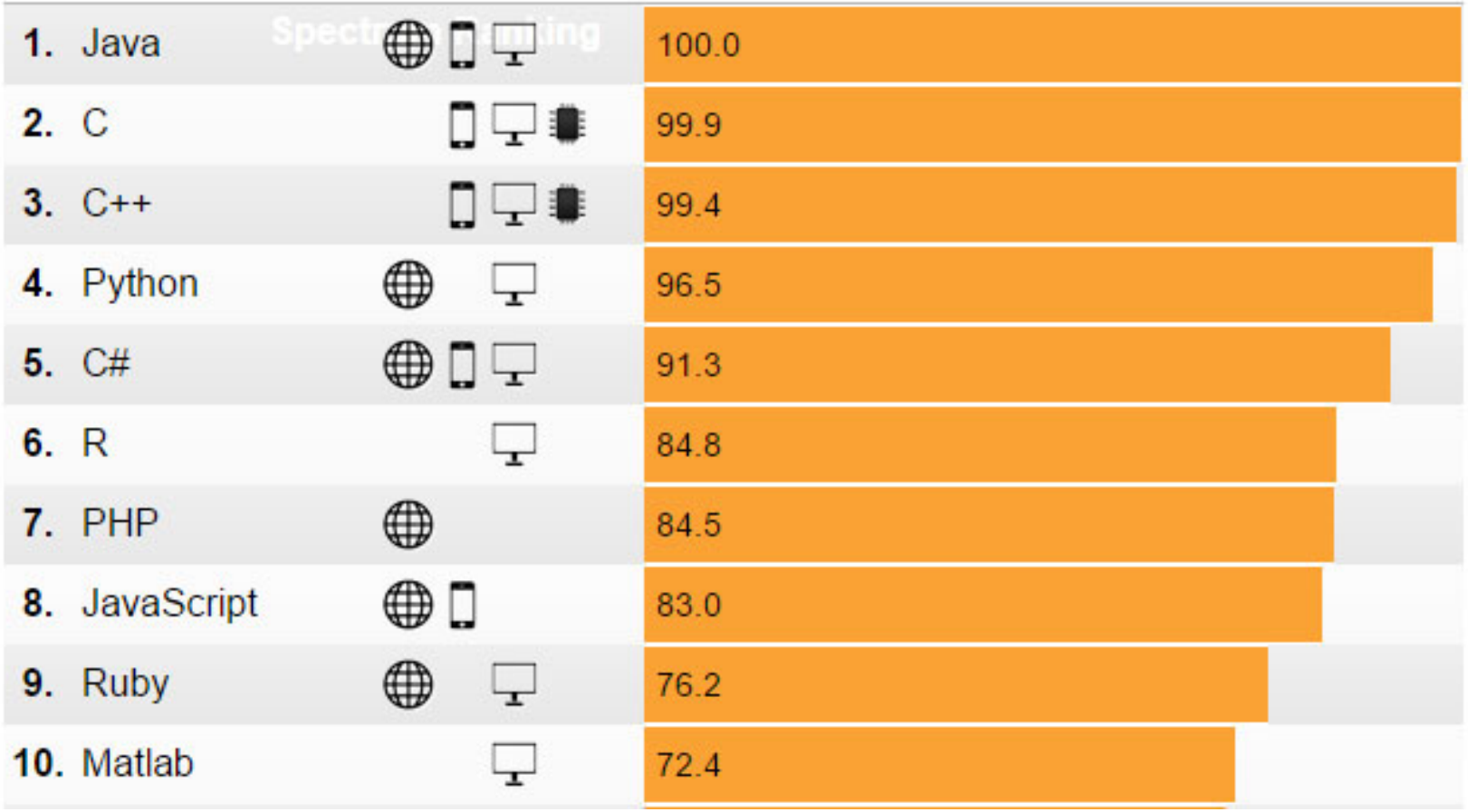
- 速く計算できるならば何でもよい
- 一般に　Fortran ＞ C/C++＞Java＞Ruby

http://www-hiraki.is.s.u-tokyo.ac.jp/lectures/prog_giho/

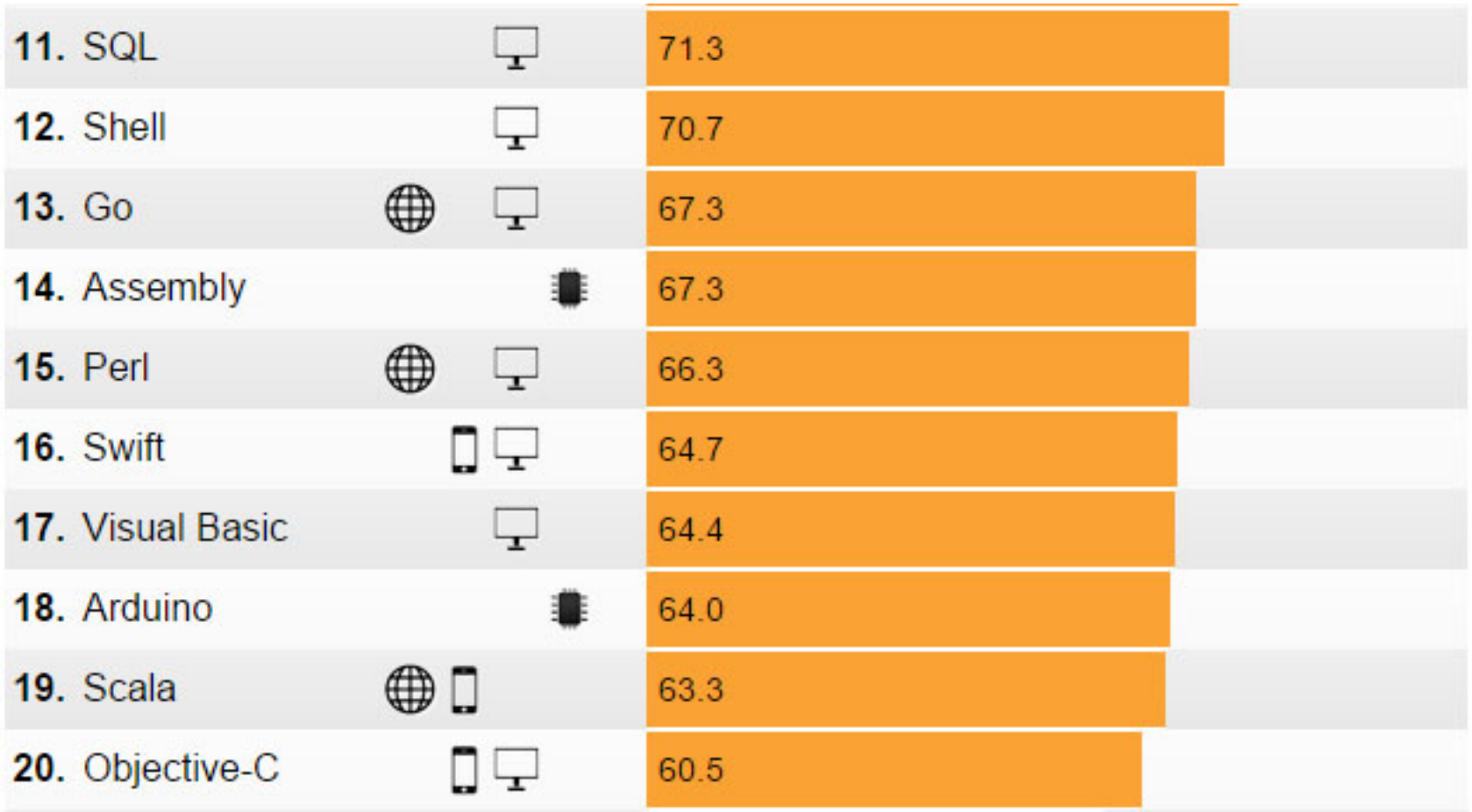
にプログラム参考例がC, Java, Rubyである。

とりあえず動かし、あとは改造して速くする努力

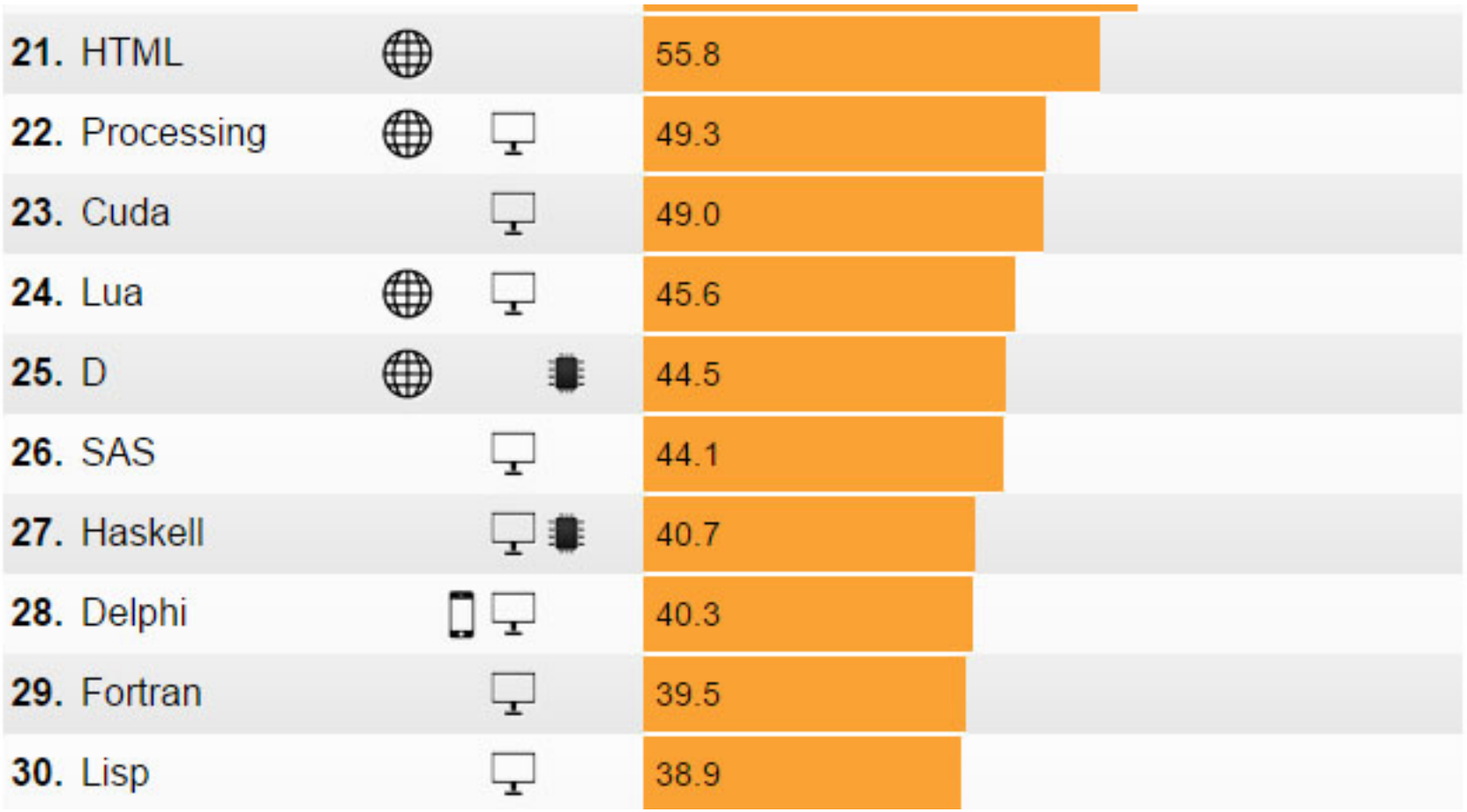
よく使われるプログラミング言語 #1~#10



よく使われるプログラミング言語 #11~#20



よく使われるプログラミング言語 #21~#30



どの言語を使うかは、第二外国語選択より難しい

時間の測り方

- 時間の測り方は多数ある
  - OSのタイマーを使う
  - CPUにあるクロックカウンタを読み取る
  - GPSから得られる時間を使う

- 本演習ではOS時間を使う

```
gettimeofday(&tv, NULL);
struct timeval {
    time_t      tv_sec;  /* 秒数       */
    suseconds_t tv_usec; /* マイクロ秒 */
}
```

時間の測り方Java、Ruby編

- Java, とても簡単

```
start_time=System.nanoTime();
```

- Ruby, とても簡単

```
start_time=Time.now.to_f()
```