Contents

1 Basic

1.1 default code

```
#ifdef LOCAL    // ========== Local ==========
void dbg() { cerr << '\n'; }
template<class T, class ...U> void dbg(T a, U ...b) {
    cerr << a << ' ', dbg(b...); }
template<class T> void org(T l, T r) { while (l != r)
    cerr << *l++ << ' '; cerr << '\n'; }
#define debug(args...) (dbg("#> (" + string(#args) + ")
    = (", args, ")"))
#define orange(args...) (cerr << "#> [" + string(#args)
    + ") = ", org(args))
#else           // ======== OnlineJudge ========
#pragma GCC optimize("O3,unroll-loops")
#pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
#define debug(...) ((void)0)
#define orange(...) ((void)0)
#endif
template<class T> bool chmin(T &a, T b) { return b < a
    and (a = b, true); }
template<class T> bool chmax(T &a, T b) { return b > a
    and (a = b, true); }
```

1.2 .vimrc

```
set nu rnu ts=4 sw=4 bs=2 ai hls cin mouse=a
color default
sy on
inoremap {<CR> {<CR>}<C-o>O
inoremap jk <Esc>
nnoremap J 5j
nnoremap K 5k
nnoremap run :w<bar>!g++ -std=c++14 -DLOCAL -Wfatal-
    errors -o test "%" && echo "done." && time ./test<
    CR>
```

1.3 Increase Stack Size (linux)

```
#include <sys/resource.h>
void increase_stack_size() {
  const rlim_t ks = 64*1024*1024;
  struct rlimit rl;
  int res=getrlimit(RLIMIT_STACK, &rl);
  if(res==0){
    if(rl.rlim_cur<ks){
      rl.rlim_cur=ks;
      res=setrlimit(RLIMIT_STACK, &rl);
} } }
```

1.4 Misc

```
編譯參數：-std=c++14 -Wall -Wshadow (-fsanitize=
    undefined)

mt19937 gen(chrono::steady_clock::now().
    time_since_epoch().count());
int randint(int lb, int ub)
{ return uniform_int_distribution<int>(lb, ub)(gen); }

#define SECs ((double)clock() / CLOCKS_PER_SEC)

struct KeyHasher {
  size_t operator()(const Key& k) const {
    return k.first + k.second * 100000;
} };
typedef unordered_map<Key,int,KeyHasher> map_t;

__builtin_popcountll    // 二進位有幾個1
__builtin_clzll         // 左起第一個1之前0的個數
__builtin_parityll      // 1的個數的奇偶性
__builtin_mul_overflow(a,b,&h) // a*b是否溢位
```