

Danh sách nhảy (Skip list)

Phát sinh ngẫu nhiên cấp của một nút

```
float frand() {
    return (float) rand() / RAND_MAX;
}
int random_level() {
    int lvl = 0;
    while ((frand() < P) && (lvl < MAX_LEVEL))
        lvl++;
    return lvl;
}
// P = 0.5
// MAX_LEVEL  $\approx \log_2 n$ 
```

Cấu trúc dữ liệu:

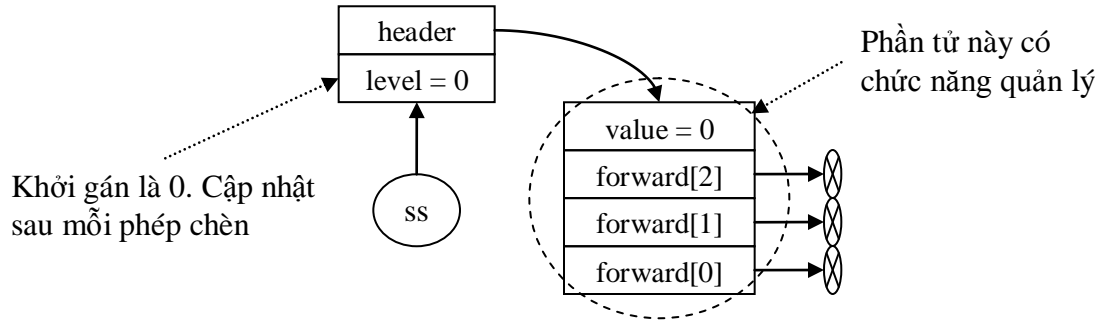
```
struct SkipNode {
    int          value;
    SkipNode **  forward;    // Mảng chứa danh sách các con trỏ, trỏ đến
                             // những nút kế (tùy theo tầng)
};
struct SkipSet {
    SkipNode *   header;
    int          level;      // Mức tối đa của một nút mà danh sách chứa
};
```

Tạo một nút

```
SkipNode * make_node(int level, int value) {
    SkipNode * sn = new SkipNode;

    sn->forward = new SkipNode * [level + 1];
    for (i = 0; i < level; i++)
        sn->forward[i] = NULL;
    sn->value = value;
    return sn;
}
SkipSet * make_skipset() {
    SkipSet * ss = new SkipSet;

    ss->header = make_node(MAX_LEVEL, 0);
    ss->level = 0;
    return ss;
}
```



Duyệt danh sách

```
void print_skipset(SkipSet * ss) {
    SkipNode * x = ss->header->forward[0];
    while (x) {
        cout << x->value << " ";
        x = x->forward[0];
    }
}
```

Tìm kiếm

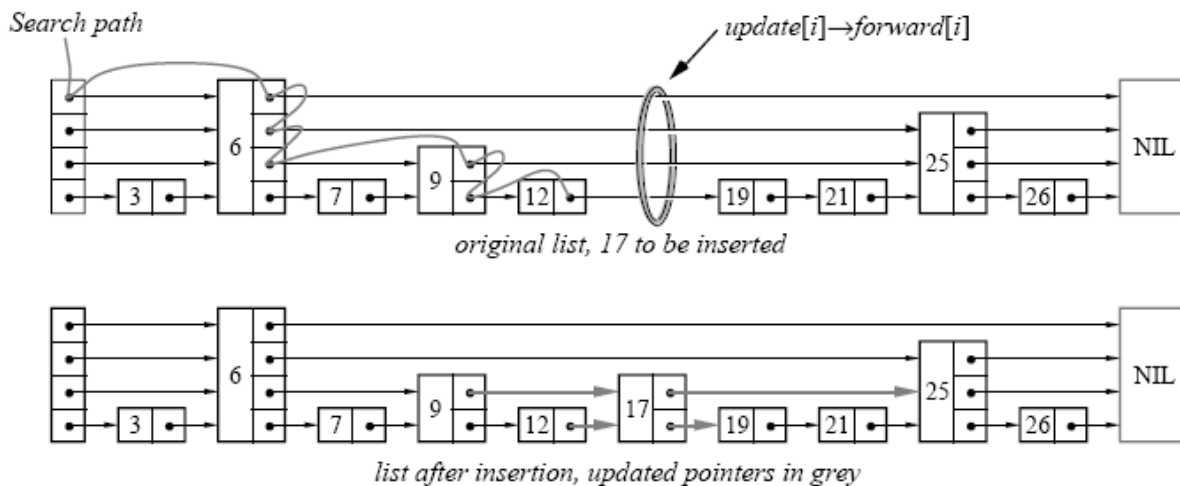
```
int search(SkipSet * ss, int k) {
    SkipNode * x = ss->header;

    for (int i = ss->level; i >= 0; i--)
        while ((x->forward[i]) && (x->forward[i]->value < k))
            x = x->forward[i];

    x = x->forward[0];
    if ((x) && (x->value == k))
        return 1;

    return 0;
}
```

Chèn nút vào danh sách



```

void insert(SkipSet * ss, int value) {
    SkipNode * x = ss->header;
    SkipNode * update[MAX_LEVEL + 1] = {NULL};

    for (int i = ss->level; i >= 0; i--) {
        while ((x->forward[i]) && (x->forward[i]->value < value))
            x = x->forward[i];
        update[i] = x;
    }

    x = x->forward[0];
    if (x == NULL || x->value != value) {
        int lvl = random_level();
        if (lvl > ss->level) { // Mức của nút mới cao nhất → cập nhật.
            for(i = ss->level + 1; i <= lvl; i++)
                update[i] = ss->header;
            ss->level = lvl;
        }
        x = make_node(lvl, value);
        for (i = 0; i <= lvl; i++) {
            x->forward[i] = update[i]->forward[i];
            update[i]->forward[i] = x;
        }
    }
}

```

Xóa một nút

```

void del(SkipSet * ss, int value) {
    SkipNode * x = ss->header;
    SkipNode * update[MAX_LEVEL + 1] = {NULL};

    for (int i = ss->level; i >= 0; i--) {
        while ((x->forward[i]) && (x->forward[i]->value < value))
            x = x->forward[i];
        update[i] = x;
    }

    x = x->forward[0];
    if (x->value == value) {
        for (i = 0; i <= ss->level; i++) {
            if (update[i]->forward[i] != x)
                break; // Thoát ra ngay ở tầng cao hơn gần nhất
            update[i]->forward[i] = x->forward[i];
        }
        delete x;

        while ((ss->level > 0) && (ss->header->forward[ss->level] == NULL))
            ss->level--;
    }
}

```