

# 応用アルゴリズム 演習

レポート執筆

# 最低限

- 学籍番号
- 名前
- 演習・課題名
- PDFを提出
  - Word/LaTeX のソフトで作成されたファイル
  - スキャンしたメモとかは NG
    - ただし、図をスキャンしてレポートに載せることは OK
- ソースコードについて
  - 基本的に別紙 (kadaiX.c とか)
  - レポートに強調したい部分だけ

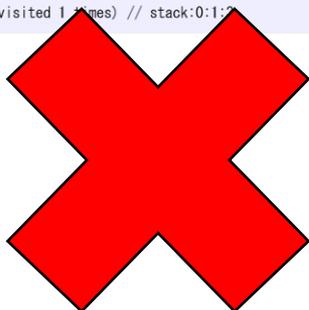
# コードの参照

- VS Code や資料のフォーマットなどを使わない
- 箱の中に,
  - モノクロ
  - フォントサイズ : 10pt
  - 等幅フォント
- TeX の場合は, `\usepackage{listing}` 等
- 全てのコードをレポートに載せることはなくて, 強調したい部分だけ
- ソースコードは別のファイルで

```
void printNode(node_t * node) {
    printf("Node ID: %d, Visited: %d, Neighbor Count: %d, Neighbors: ",
        node->id, node->visited, node->neighborCount);
    for (int i = 0; i < node->neighborCount; i++) {
        printf("%d ", node->neighbors[i]->id);
    }
    printf("\n");
}
```

問題 1

```
(node 0, visited 0 times) // stack:0
(node 2, visited 0 times) // stack:0:2
(node 3, visited 0 times) // stack:0:2:3
(node 0, visited 1 times) // stack:0:2:3:0
(node 1, visited 0 times) // stack:0:1
(node 2, visited 1 times) // stack:0:1:2
(node 3, visited 1 times) // stack:0:1:2:3
```



例

```
void dfs(node_t * node) {
    printNode(node);
    node->visited++;
    if (node->visited == 1) {
        int n;
        for (n = 0; n < node->neighborCount; n++) {
            node_t * next = node->neighbors[n];
            dfs(next);
        }
    }
}
```