

# Homework 0

Simon Zheng  
260744353

*MONTH DAY*<sup>SUFFIX</sup>, *YEAR*

## 1

Algorithm: BDFS(Vertex  $u$ )

---

1	Input: A vertex $u$ <i>in</i> a graph $G$
2	Output:

---

## 2 Master Theorem

## 3 Stock

Recursively divide the sequence in half. When merging, keep track of the minimum price found and its index and take the differences between each pair of merged prices. Also keep track of a global maximal difference found.

## 4 Huffman code tree

## 5 Compression scheme

By the pigeonhole principle.

We have  $256^n = 2^{8n}$  possible strings for an  $n$ -character file (ASCII is 8-bit which gives us 256 distinct characters).

We have an  $8n$ -character file that we try to compress using  $C$ . When compressing we want to be able to reliably decompress it so we need a unique mapping for every distinct file (combination of string).

We need  $2^{8n}$  bits to represent  $8n$  distinct characters. But these characters can be arranged into  $2^{8n}$  distinct ways. The best  $C$  can do is getting down to  $2^n$  bytes.

Thus, the actual number of possibilities  $C$  must handle is from  $2^n$  to  $2^{8n}$ , including the case where it compresses down to  $2^n$  bytes. This means  $2^{8n} - 2^n + 2^{n-1}$  (because the minimum/best case  $2^n$  is included).

Therefore, using the pigeonhole principle, since a compression algorithm is limited by the number of characters in this way, it cannot possibly do better than  $8n$ .

## 6 Tile board

Keep recursively dividing the board into 4 smaller and smaller "quadrants" (as it is of size  $2^n \times 2^n$ ) until you get to single tiles.

We will start from the missing cell (a single tile).

First, "merge" back up one level so you get  $2 \times 2$  quadrants. If a quadrant contains the empty cell, then there must be 3 empty cells in an L shape, so put a tile on it.

Now, together with the 3 other quadrants (that will form a bigger quadrant in the "upper" recursion level), put a tile in the center of all 4. Since the quadrant containing the empty cell is filled, the center 4 cells will have one filled cell, and the 3 others will be empty and in an L shape, just like previously, so fill them with a tile.

Doing this means that the 3 other quadrants (that didn't contain the empty cell) are going to have one cell filled, and will each have 3 empty cells left in an L shape, so fill them with tiles.

Keep doing this recursively, where in every recursion layer you always start by filling the center 4 cells with a tile, as there will be a single cell filled previously in a lower recursion level.

This works by induction.

If  $2^n \times 2^n$  is the dimension of the board:

Base case  $P(n) = P(1)$ : We have a  $2 \times 2$  square board. Remove any of the cells. You will be left with 3 cells in an L shape.

Induction hypothesis: For any  $2^n \times 2^n$  square board with a single missing cell, you can fill it with L-shaped 3-cell tiles.

Induction step  $P(n + 1)$ : If the previous board  $P(n)$  is part of a bigger  $2^n \times 2^n$  board, then it must be with 3 other same size board in a quadrant fashion (to form a square). We know that they are all empty except for the "sub-board" we previously filled. Therefore, at the center of the "super-board", there must be 4 cells which are the corners of each sub-board that touches each other, with one that filled or has the missing cell. We can put a tile on the 3 others, and as such the 3 other sub-board will each have a single cell filled. Thus, to fill the 3 other boards, we do the same thing as the first as it is the same case, just that the newly filled cells act like the missing cell in the first sub-board.

## 7 Triple cycles

Make an adjacency matrix. One side is outgoing, other incoming. But this doesn't matter.

Cube it to get paths of 3. The diagonal is where paths came back to the original vertex of that column/row, so just sum it (get the trace) to get all triple cycles.

If  $a \rightarrow b \rightarrow c \rightarrow a$  and  $b \rightarrow c \rightarrow a \rightarrow b$  are considered the same triple, then divide the trace by 3.