

DHomework 1

Carter Owens

1

Standard-cell: improves design time but sacrifices flexibility and density.

Custom: extreme flexibility and density at the cost of very high design time.

2

the incidence matrix

$$B = \begin{bmatrix} -1 & -1 & 0 & 0 \\ 1 & 0 & 1 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

when multiplied by its transpose: BB^T results in the following matrix

$$X = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

it appears that each column and each row (since the matrix is symmetrical) indicates the connections of each row. the entry $X_{(i,j)}$ will contain -1 if nodes i and j are connected, the direction of the connection is lost however, and if i and j are equal (the diagonal of the matrix) the matrix entry contains the number of connections that node has. essentially it becomes the negative adjacency matrix with the addition of the diagonal containing the number of connections.

3

$$n \quad n \log^2 n \quad n\sqrt{n} \quad n^2 \quad n^{\log n} \quad n^2 \log n \quad 2^n \quad n^{0.5n} \quad n^9$$

this order was found by evaluating each expression at increasingly high values of n, the higher the result the more complex

4

a

$$n \log n$$

b

$$n^2$$

c

$$n \log_e n$$

d

$$e^n$$

5

a

false: for example if

$$O(f(n)) = n^2 O(g(n)) = n^3$$

then

$$O(f(n)) = O(g(n))$$

would still technically be true, even though a f is a simpler algorithm it will still be complete in n^3 time, but $O(f(n)) = O(g(n))$ would be false since g would not be complete in n^3 time

b

true: the order of the statements implies that either the complexities are equal. or that h is the most complex. and since all the equalities are going the same way, there is on requirement that the algorithms must be equally complex. the insinuation is essentially: $f \leq g$ and $g \leq h$ therefore the statement $f \leq h$ holds