Question 1

a) $f(n) = n^3$ $g(n) = (n+2)^3 =$ $n^3 +$ lower order terms.

$f(n) \in \Theta(g(n))$

---

b). $f(n) = 3^{(2n)}$ $= (3^2)^n$.

$g(n) = 3^{(n+2)}$ $= c \cdot 3^n$

$\Rightarrow$ $f(n)$ has bigger base.

$\Rightarrow$. $f(n) \in \Omega(g(n))$.

---

c). $f(n) = (3n)^2 = c \cdot n^2 \to$ Polynomial.

$g(n) = 3^{n^2}$. $= 3^{2n}$ $\to$. Exp.

$g(n) > f(n)$.

$f(n) \in O(g(n))$.

---

d). ~~f(n) =~~

$f(n) = (\log_2(n))^{\log_2(n)}$ $\to$ 2^x for some large n.

$\to$ exponent.

$g(n) = \sqrt{n}$. $\to$ ~~Logarithms~~. root.

$\Rightarrow$. $f(n) \in O(g(n))$.

Question 2

a). $\sum_{i=0}^{n-2} \cancel{4} \cdot \sum_{i+1}^{n-1}$ $\Rightarrow$ triangle numbers backwards ...?

c

$= (n-2)\left(\frac{(n-1)(n-1+1)}{2}\right)$.

t

$= (n-2)\left(\frac{(n-1)n}{2}\right)$.

$= (2n-4)(n^2-n)$.

$= \cancel{2n^3+4}\ 2n^3 - 2n^2 - 4n^2 + 4n$.

$\Rightarrow c \cdot n^3 +$ lower order terms.

$\therefore \Theta(n^3)$.

b). The above algorithm calculates

Question 3

a). DFS from node A.

$A \xrightarrow{4} C \xrightarrow{3} D \xrightarrow{1} F \xrightarrow{12} G \xrightarrow{9} E \xrightarrow{7} B$.

b). BFS. from A.

$A \xrightarrow{4}$ ~~C -> D -> B. -> F -> E. -> G.~~ A→C D B F E→G.

c). cost (DFS) = 4 + 3 + 1 + 12 + 9 + 7 = 36.

cost (BFS) = 4 + 5 + 8 + 1 + 7 + 9 = 34.

∴ BFS result should be used.

d). function FindPathThanksToDijkstra (Graph, source) ~~[illegible]~~.

```
Q ← empty set of vertices.
foreach vertex v in Graph: do.
  dist[v] ← ∞
  prev[v] ← Null.
  push v onto Q.
dist[source] = 0.
while Q not empty do
  u ← ~~vertex~~ minDist[u] in Q.
  pop (u)
  foreach v where (u, v) do.
    ~~dist ← dist[u,v] + |u+v|~~
    distance ← |(u,v)| + dist[u].
    if distance < dist[v].
      dist[v] ← distance.
      prev[v] ← u.
```

return prev

Q4.

a) ~~Changing base.~~

~~$T(n) = T(\frac{n}{3}) + T(\frac{2n}{3})$.~~

Worst case of binary search.

log(n) where element doesn't exist.

modifying the split of binary Search

only adds a constant factor in

worst case so that $\Rightarrow bs' \in \Theta(\log(n))$.

b). $T(n) = (\frac{n}{5}) + 1.$

$= \frac{1}{5}(n) + 1.$

$= \frac{1}{5^2}(n) + 1 + 1.$

$\vdots$

$= T(5^{-k}(n)) + k.$

$= 5^{-(n-1)}(n) + (n-1)$

$=$ ~~...~~ ;