1. Consider the weighted dag $G$ shown below.

Solve the single source minimum path problem for $G$, where $V_0$ is the start vertex. Your answer should consist of two arrays, one which shows the minimum cost of a path to each $V_i$, and the other of which gives the back pointers.

2. Consider the function $\text{george}(\text{int } n)$ computed by the following recursive C++ code.

```cpp
int george(int n)
{
    assert(n >= 0);
    if (n < 2) return 1;
    else return 2*george((n+1)/2)+2*george(n/2)+2*george((n-1)/2)+2*george(n/2-1);
}
```

(a) Use the master theorem to compute the time complexity of the above code.
(b) Design a dynamic programming algorithm to compute $\text{george}(n)$ for any given $n$. You do not have to write C++ code. The time complexity of your algorithm should be $\Theta(n)$.

(c) Design a memoization algorithm to compute $\text{george}(n)$ for any given $n$. You do not have to write C++ code. This algorithm should be much faster than the dynamic programming algorithm. What is its time complexity?

3. Write pseudocode for the Floyd-Warshall Algorithm.