Report for Assignment 4

Deep Kiran Shroti
Roll Number- 12EC35013

March 9, 2016

Last man standing

Assignment Statement

- We start with n people numbered 1 to n around a circle
- We eliminate every second remaining person until only one survives

Task is to compute the position of the survivor as V(n).
Write a program to generate the elimination order according to the given scheme, but preferably without using an array (or linked list), and thereby determine the survivor.

Check both programmatically and analytically whether the survivor position can be obtained as: V(2_m+ l) = 2l + 1, for $m \geq 0$ and $0 \leq l < 2m$.

1 Implementation Details

We can find the last man standing with the help of recurrence.

1. **First Step:** The number of men standing is taken from the user as input and the generated series is passed on to the function *lastManStanding()* function.
2. **Second Step:** In the *lastManStanding()* function the every alternate number is printed, i.e. the series of man that will be eliminated.
3. **Third Step:** The remaining man that are left are passed on to a *leftMan-Standing()* function for further round of elimination.

2 Analysis

Proof: We use strong induction on n. The base case n=1 is true. We consider separately the cases when n is even and when n is odd.

If n is even, then choose l_1 and m_1 such that $n/2 = 2^{m_1} + l_1$ and $0 \leq l_1 < 2^{m_1}$. Note that $l_1 = l/2$. We have $f(n) = 2f(n/2) - 1 = 2((2l_1) + 1) - 1 = 2l + 1$, where the second equality follows from the induction hypothesis.

If n is odd, then choose l_1 and m_1 such that $(n-1)/2 = 2^{m_1} + l_1$ and $0 \leq l_1 < 2^{m_1}$. Note that $l_1 = (l-1)/2$. We have $f(n) = 2f((n-1)/2) + 1 = 2((2l_1) + 1) + 1 = 2l + 1$, where the second equality follows from the induction hypothesis. This completes the proof.

3 Conclusion

In the algorithm the *lastManStanding()* function divides the one series into a series of size half.

$$T(n) = T(n/2) + b$$

Therefore the complexity is O(*logn*).