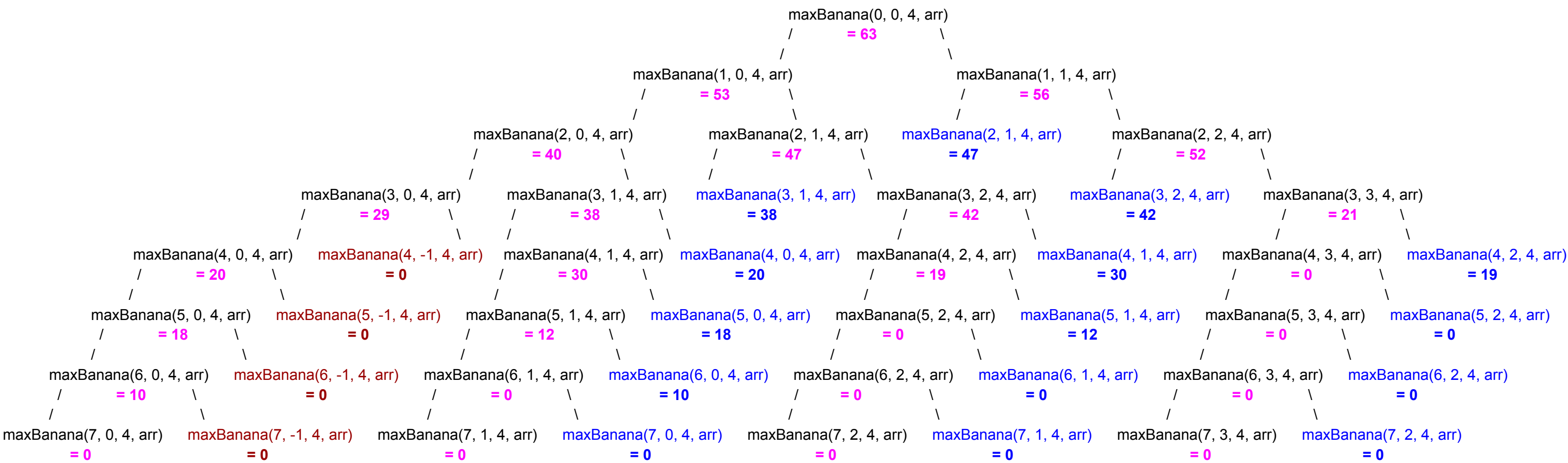


Recursive tree of LightOJ Monkey Banana Problem



**Note:** Black color indicates recursive calls  
Magenta color indicates results obtained from recursive call  
Blue color indicates results obtained from memoization  
Red berry color indicates results obtained from corner cases.

The recursive function is:

```
long long maxBanana(int row, int col, int n, const vector<vector<int>>& arr)
{
    if (row == 2 * n - 1) [corner case]
    {
        return 0;
    }
}
```

```
}

if (col < 0)                                [corner case]
{
    return 0;
}

if (dp[row][col] != -1)
{
    return dp[row][col];
}

long long ans = arr[row][col];

if (row < n - 1)
{
    ans += max(maxBanana(row + 1, col, n, arr), maxBanana(row + 1, col + 1, n, arr));    [recursive equation]
}
else
{
    ans += max(maxBanana(row + 1, col, n, arr), maxBanana(row + 1, col - 1, n, arr));    [recursive equation]
}

dp[row][col] = ans;
return ans;
}
```