

UM-20

MASS CURVE ANALYSIS AND SEQUENT PEAK ALGORITHM

SATISH CHANDRA

DIRECTOR

STUDY GROUP

S.K. JAIN

National Institute of Hydrology

Jal Vigyan Bhavan

Roorkee - 247667

1985 - 86

CONTENTS

| | PAGE |
|----------------------------|------|
| Abstract | 1 |
| 1.0 INTRODUCTION | 2 |
| 2.0 MASS CURVE TECHNIQUE | 3 |
| 3.0 SEQUENT PEAK ALGORITHM | 7 |
| 4.0 CONCLUSIONS | 11 |
| REFERENCES | |
| APPENDIX | |

ABSTRACT

The computation of required storage is one of the important steps of reservoir design. Two computer programmes have been described in this report which make use of mass curve method and sequent peak algorithm for this purpose. As mass curve is basically a graphical technique, computer graphics has been used for the development of programme of mass curve method. Both the programmes are interactive in nature.

1.0 INTRODUCTION

The storage capacity to be provided for a particular reservoir is one of the most important decision associated with reservoir design. The maximum storage capacity which could be provided is limited because of water availability and the characteristics of site. But as the reservoir projects are highly capital intensive, it is imperative that the storage capacity is just sufficient consistent with the demands.

1.1 Techniques for Reservoir Capacity Computation

A large number of techniques are available for reservoir capacity computation and these can be subdivided into four major classes:

1. Critical period techniques
2. Stochastic techniques
3. Simulation techniques
4. Optimization techniques

Among all the techniques, those based on the concept of critical period were developed first. The critical period is defined as the duration in which an initially full reservoir depletes and passing through various states (without spilling) empties. In the methods based on critical period concept, a sequence of streamflows containing a critical period is passed through an initially full reservoir while releases are made to meet the demands. The reservoir capacity required is equated to the maximum difference between cumulative inflows and cumulative releases.

1.2 Scope of the Present Report

In the present report two computer programmes have been described which can be used to calculate the storage capacity required in a

reservoir using either mass curve or sequent peak algorithm.

2.0 MASS CURVE TECHNIQUE

The mass curve technique, proposed by Ripple in 1883 to determine storage capacity of a reservoir, is a graphical integration technique.

Let x be the series of inflows to the reservoir and q be the outflow or draft series and Z be defined as

$$\begin{aligned} Z_t &= \int_0^t (x - q) d\tau \\ &= \int_0^t x d\tau - \int_0^t q d\tau \end{aligned}$$

or $Z_t = X_t - Q_t \quad \dots (1)$

The plot of Z_t with respect to time represents storage fluctuations in an unconstrained reservoir subject to inflow x and outflow q . This graph can be used to find the smallest size of the reservoir required to supply draft series Q_t throughout the critical period without failure. Here it is assumed that the reservoir does not spill during the critical period or it is a topless reservoir. It may be mentioned that Ripple assumed reservoir to be empty at the start of critical period. This assumption is more realistic than the one assumed in one other variation of mass curve technique where the reservoir is assumed full in the beginning. This is particularly true for large reservoirs.

Using the mass curve technique graphically, the computations are performed as given in following steps:

- i) First the mass curve of cumulative inflow relative to draft is plotted.

- ii) Now horizontal lines from the troughs of this mass curve are drawn.
- iii) The required reservoir capacity is given by the maximum intercept between the mass curve and the horizontal lines. This will be the size of smallest reservoir which will be able to supply desired draft without failure during the time period under consideration. Sometimes, it is suggested to plot cumulative inflow curve and then plot cumulative draft curve and then measure the intercept after drawing tangent from the mass curve. A typical of such graphs is shown in figure 1. But by plotting cumulative inflow less draft the procedure is made very simple and variable demands can be easily taken care of.

Many times, it is more convenient to express release as a ratio of mean inflow and this ratio is called the degree of regulation or release ratio. Similarly, the storage capacity can also be expressed as a ratio of mean annual inflow and is called storage ratio or storage coefficient.

The mass curve technique, although very simple and straight forward, has few shortcomings. One drawback pointed out many times is the implicit assumption that the storage which would have been adequate in past will also be adequate in future. Although this is not clearly true, the error caused is not really serious particularly if sufficiently long flow series has been considered. Secondly this problem will arise in any other method since true future is not known. Some methods try to address this problem by explicitly considering the stochasticity of the inflows.

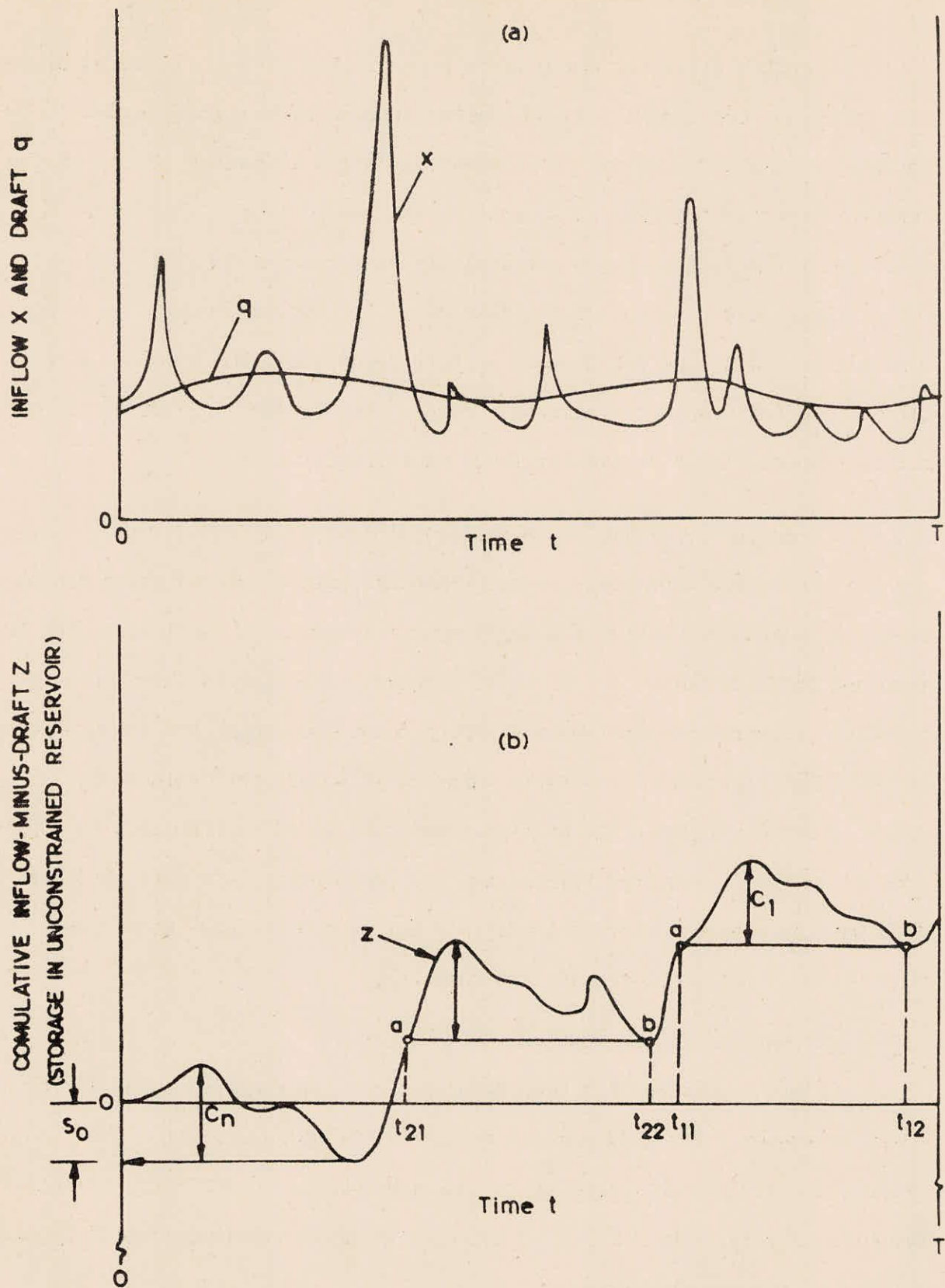


FIG. 1 Determination of reservoir storage capacity using mass curve method

Another criticism which is in general true for all critical period techniques is the circularity of the definition of critical period. By definition, the critical period depends upon the operating policy of the reservoir up to that period, its capacity and demand level. These are essentially the factors which our analysis aims to find out.

One more drawback of the mass curve is that explicit economic analysis can not be done in this technique. The storage size can not be related to the economic life of the project. Further, it can not be computed for a particular level of reliability.

2.1 Computer Programme for Mass Curve Analysis

An interactive computer programme has been developed which can be used for mass curve analysis using computer graphics. The inflow data is read and the mass curve of inflows is plotted. The user is then asked to input the required release and a starting point from which the computations start. The capacity of reservoir which could supply this release is computed and displayed. The user is then again asked (optionally) to supply another value of required release and the computations are performed again. This way the user can carry the analysis for as many release values as desired and finally the results are tabulated.

2.2 Programme Implementation

This programme has been developed and implemented on VAX 11/780 computer system. The programme is written in FORTRAN language. The graphical part is taken care of by the subroutines of PLOT-10 Interactive Graphics Library. The TEK-4027-A coloured graphics terminal from Tektronix is used to display the results.

2.3 Description of the Programme

The main programme is written in FORTRAN language. It reads data from the logical unit one. The main programme calls a number of IGL sub-routines for various graphical applications. The programme interactively prompts for the inputs from the user and displays the results as requested. A listing of the programme is given in Appendix 1 and a sample input data in appendix 2.

2.4 Programme Input

The input to the programme is read from the logical unit no. 1 whose structure has to be as follows:

| Line No. | Variable(s) | Format | Description |
|----------|-------------|--------|----------------------------|
| 1 | N | Free | No. of periods of analysis |
| 2...N+1 | T and FL | Free | Time and flow |

Further input is supplied interactively at the time of running the programme.

3.0 SEQUENT PEAK ALGORITHM

This method was proposed as a method which circumvents the need to choose the correct starting storage which is required in the mass curve procedure. The storage required to supply a given demand is the range of the cumulative inflow less draft. The computations are quite simple and can be carried out as follows. Let I_t be the inflow to the reservoir in the period t , R_t be the release from the reservoir, and S_t the storage at the beginning of the period t . The reservoir is assumed to be empty in the beginning. The computations are performed for twice the length of the inflow record. It is assumed that the inflows repeat

after the end of first cycle. This assumption is made to take care of the case when the critical period falls at the end of the record. The variable S_t is calculated by the following equation:

$$S_t = \begin{cases} S_{t-1} + R_t - I_t & \text{if positive} \\ 0 & \text{if negative or zero} \end{cases} \quad \dots (2)$$

The required storage capacity is equal to the maximum of S_t values. The computations are illustrated in the following table for a simple case where it is required to release 5 units of water from the reservoir and the inflows are as given in column 4 of the following table:

| Period | Storage required in previous period | Release | Inflow | Storage needed in the present per |
|--------|--|---------|---------|---|
| t | $[S_{t-1}]$ | + R_t | - I_t | = S_t |
| 1 | 0 | 5 | 3 | 0 |
| 2 | 0 | 5 | 4 | 1 |
| 3 | 1 | 5 | 4 | 2 |
| 4 | 2 | 5 | 1 | 6 |
| 5 | 6 | 5 | 2 | 9 |
| 6 | 9 | 5 | 2 | 12 |
| 7 | 12 | 5 | 6 | 11 |
| 8 | 11 | 5 | 5 | 11 |
| 9 | 11 | 5 | 8 | 8 |
| 10 | 8 | 5 | 9 | 4 |
| 11 | 4 | 5 | 7 | 2 |
| 12 | 2 | 5 | 7 | 0 |

Thus the required reservoir capacity would be 12 units which is the maximum of the last column in the above table. Here the calculations are not repeated for the second cycle because the storage at the end of first cycle is zero and therefore the second cycle would be identical to the first cycle.

3.1 Properties & Limitations

The sequent peak algorithm can very easily consider the variable release from the reservoir. The reliability of the reservoir can be obtained indirectly. Since the reservoir would be able to meet the worst drought from the record, the implied probability of failure would be $1/(N+1)$. The algorithm is very fast and easy to programme.

A single historical record is used to compute the storage and hence the method is limited in that sense. It is also not possible to exactly consider the losses, they can be approximately included in the releases.

3.2 Description of Programme

The main programme which carries out all computations is a small programme written in VAX-11 FORTRAN language. The programme reads the input data from logical unit one. The results are also typed from the main programme. No subprogrammes are required for successful running of the programme.

3.3 Programme Implementation

This programme has been developed and tested on VAX 11/780 computer system with FORTRAN 77 compiler. The programme is sufficiently generalized. The statements which may require change for implementation of this programme on other systems are the OPEN statement and the READ statements. BLOCK IF statement has also been used in this programme which is not supported by many compilers.

3.4 Description of Variables

Following is the list of important variables with description which are needed for preparing input data for this programme.

| <u>Variable</u> | Description |
|-----------------|--|
| N | Number of periods for which data is available |
| FLOW | Inflow to reservoir during a particular period |
| IDO | A flag which equals 1 if detailed output is required |
| IC | A flag which equals 1 if the demands are fixed and zero if they vary with time |
| RELE | The constant volume of water which is to be released from the reservoir. |
| REL | The volume of water to be released from the reservoir during a period |
| SREQ | The storage capacity required. |

3.5 Programme Input

The input data is read from logical unit number one. The corresponding file declared in this case is SEQ.DAT. The input data may be in free format. The organisation of input file should be as follows:

| Line No(s) | Variable | Format | Description |
|-------------|-------------|--------|--|
| 1 | N | A | Title of the problem (max 80 characters long) |
| 2 | N | Free | No. of periods for which analysis has to be performed. |
| | IC | Free | A flag which equals one if only one value of constant demand is to be read |
| | IDO | Free | Should be equal 1 if table of detailed computations is required |
| 3 onwards | FLOW | Free | Inflows to the reservoir during each period |
| As required | RELE or REL | Free | Constant or varying demands depending upon IC. |

The input data can be in any consistent system of units.

3.6 Programame Output

The output from the program consists of Title of the problem and the basic parameters of the problem. A table showing stepwise computations is printed if requested. i.e., IDO = 1. Finally the programmes types the required storage capacity.

A listing of the programame alongwith sample input and output is given in appendix 3, 4, and 5 respectively.

4.0 CONCLUSIONS

Two computer programames developed for determining the storage capacity of a reservoir using the mass curve method and the sequent peak algorithm have been developed and are described in this report. Since the mass curve method is essentially a graphical method, the programme for this method has been developed to work in a graphaical environment.

REFERENCES

1. Fiering, M.B., (1967), Streamflow Synthesis, MacMillan Publishing Company, London.
2. Loucks, DP, JR Stedinger and DA Haith, (1981) Water Resources Systems Planning and Analysis, Prentice-Hall Inc., Englewood Cliffs, New Jersey.
3. McMohan, T.A., and R.G. Mein, (1968) Reservoir Capacity and Yield, Developments in Water Science - Elsevier Scientific Publishing Company, New York.

```

C *****
C PROGRAM FOR RESERVOIR CAPACITY COMPUTATION USING MASS CURVE
C CALLS PLOT-10 IGL ROUTINES FOR GRAPHICAL TASKS
C *****
C DIMENSION T(50),FL(50),CFL(50),XCOR(3),YCOR(3),X(20),cfn(50)
1,rel(30),GAP(20),XT(50),YT(50),Y(20)
LOGICAL COND
OPEN(UNIT=1,FILE='FLOW.DAT',STATUS='OLD')
DATA XCOR/ 20,20,128/
DATA YCOR/100,20, 20/
READ(1,*) N
5 DO 5 I=1,N
READ(1,*) T(I),FL(I)
CFL(1)=FL(1)
DO 20 I=2,N
20 CFL(I)=FL(I)+CFL(I-1)
FX=(T(N)-0.0)/100.0
FY=(CFL(N)-0.0)/70.0
X(1)=0
Y(1)=0
DO 30 I=2,11
30 Y(I)=Y(I-1)+FY*10.0
X(I)=X(I-1)+FX*10.0
DO 220 I=1,N
220 XT(I)=T(I)/FX+20.0
YT(I)=CFL(I)/FY+20.0
BT=T(2)-T(1)
CALL GRSTRT(4027,1)
CALL TXGAP(-99.,-99.)
CALL TXICUR(5)
CALL SKIP
CALL POLY(3,XCOR,YCOR)
C FOR POINTS ON Y AXIS
DO 60 I=1,12,2
60 AI=I*10.+10.

```

```

CALL MOVE(20.,AI)
CALL VECREL
CALL MOVE(-2.,0.)
CALL DRAW(2.,0.)
CALL MOVE(-10.,0.)
CALL VECABS
60 CALL RNUMBR(Y(I),1,6)
C FOR POINTS ON X AXIS
DO 45 I=1,12,2 ,
AI=I*10.+10.
CALL MOVE(AI,20.)
CALL VECREL
CALL MOVE(0.,-2.)
CALL DRAW(0.,2.)
CALL MOVE(-2.,-5.)
CALL VECABS
45 CALL RNUMBR(X(I),1,4)
CALL MOVE(20.,20.)
DO 25 I=1,N
25 CALL DRAW(XT(I),YT(I))
CALL TXAM
CALL TXANGL(0.001)
CALL MOVE(70.0,10.0)
CALL TEXT(16,'TIME (HRS) ---->')
CALL TXANGL(90.)
CALL MOVE(2.0,40.0)
CALL TEXT(25,'CUMULATIVE DISCHARGE --->')
J=1
8 CALL GETURN(33,'PLEASE GIVE RELEASE PER PERIOD :',6,REE,LR)
REL(J)=REE
CALL MOVE(XLOC,YLOC)
type *, ' PLEASE GIVE THE STARTING POINT'
CALL LOCATE(1,XLOC,YLOC,IDAT,IG2)
XLO=(XLOC-20.0)*FX
IXL=XLO
CALL MARKER(XLOC,YLOC,13)
R2=REE*(T(N)-IXLO)
YFM=R2/FY+YLOC
CALL MOVE(XLOC,YLOC)
CALL DRAW(XT(N),YFM)

```

```

COND=,FALSE,
CFN(IXL+1)=FL(IXL)*(XLO-IXL)
DO 58 I=ixl+1,N
AA=REE*(T(I)-XLO)
GAPJ=AA-CFN(I)
IF(GAPJ.GT,GAP(J)) GAP(J)=GAPJ
IF(AA.LT,CFN(I)) COND=,TRUE,
CFN(I+1)=CFN(I)+FL(I+1)
58 CONTINUE
IF(GAP(J).LT,0) GAP(J)=0
IF(,NOT,COND) GAP(J)=-999
CALL GETUTX(29,' DO YOU WANT TO CONTINUE ? : ',1,IRES,LL)
IF(IRES.EQ,'Y',or,ires.eq,'y') then
J=J+1
CALL MOVE(XLOC,YLOC)
CALL remove
CALL DRAW(XT(N),YFM)
CALL APPEAR
GO TO 8
endif
CALL GRSTOP
TYPE *,'!WOR 0'
TYPE *,' '
TYPE *,' '
TYPE *,' I RELEASE REQUIRED STORAGE'
TYPE *,' '
TYPE 14, (I,REL(I),GAP(I),I=1,J)
14 FORMAT(7X,I4,2F12.3/)
STOP
END

```


FLOW.DAT

| | |
|----|---|
| 16 | |
| 1 | 3 |
| 2 | 4 |
| 3 | 4 |
| 4 | 1 |
| 5 | 2 |
| 6 | 2 |
| 7 | 6 |
| 8 | 5 |
| 9 | 8 |
| 10 | 9 |
| 11 | 7 |
| 12 | 7 |
| 13 | 2 |
| 14 | 4 |
| 15 | 5 |
| 16 | 2 |

```

C      RESERVOIR CAPACITY COMPUTATION USING SEQUENT PEAK ALGORITHM
      DIMENSION FLOW(500),STOR(500),REL(500)
      CHARACTER*80 TITLE
      OPEN(UNIT=1,FILE='SEQ.DAT',STATUS='OLD')
***** GET INPUT
      READ(1,5) TITLE
5      FORMAT(A)
      READ(1,*) N,IC,IDO
      READ(1,*) (FLOW(I),I=1,N)
      IF(IC.EQ.1) THEN
        READ(1,*) RELE
        DO 10 I=1,N
10       REL(I)=RELE
        GO TO 1
      ENDIF
      READ(1,*) (REL(I),I=1,N)
***** READING OVER
1      CONTINUE
      N2=N+N
      DO 20 I=1,N2
        FLOW(N+I)=FLOW(I)
20       REL(N+I)=REL(I)
        SREQ=STOR(1)
        DO 100 I=1,N2
          STOR(I+1)=STOR(I)+REL(I)-FLOW(I)
          IF(STOR(I+1).LT.0) STOR(I+1)=0
          IF(STOR(I+1).GT.SREQ) SREQ=STOR(I+1)
100      CONTINUE
***** OUTPUT THE RESULTS
      TYPE 5, TITLE
      TYPE 25, N,2*N
25      FORMAT('      Input data available for 'i3' periods. '//
1         '      Analysis carried for 'i3' periods, i.e., '//
2         '      twice the length of the record. '//)
      IF(IDO.EQ.1) THEN
        TYPE *, ' PERIOD          S(T)          INF(T)          REL(T)
1         S(T+1) '
        DO I=1,N2
          TYPE 16,I,STOR(I),FLOW(I),REL(I),STOR(I+1)
        END DO
16      FORMAT(4X,I3,4F15.2)
      ENDIF
      TYPE 15, SREQ
15      FORMAT('///      THE REQUIRED STORAGE IS : 'F10.2)
      STOP
      END

```

Reservoir capacity using sequent peak method
 14 1 1
 5 7 8 4 3 3 2 1 3 6 8 9 3 4 9
 4.5
 3 4 4 1 2 2 6 5 8 9 7 7 2 4 5 2

Reservoir capacity using sequent peak method
 Input data available for 14 periods.

Analysis carried for 28 periods, i.e.,
 twice the length of the record.

| PERIOD | S(T) | INF(T) | REL(T) | S(T+1) |
|--------|-------|--------|--------|--------|
| 1 | 0.00 | 5.00 | 4.50 | 0.00 |
| 2 | 0.00 | 7.00 | 4.50 | 0.00 |
| 3 | 0.00 | 8.00 | 4.50 | 0.00 |
| 4 | 0.00 | 4.00 | 4.50 | 0.50 |
| 5 | 0.50 | 3.00 | 4.50 | 2.00 |
| 6 | 2.00 | 3.00 | 4.50 | 3.50 |
| 7 | 3.50 | 2.00 | 4.50 | 6.00 |
| 8 | 6.00 | 1.00 | 4.50 | 9.50 |
| 9 | 9.50 | 3.00 | 4.50 | 11.00 |
| 10 | 11.00 | 6.00 | 4.50 | 9.50 |
| 11 | 9.50 | 8.00 | 4.50 | 6.00 |
| 12 | 6.00 | 9.00 | 4.50 | 1.50 |
| 13 | 1.50 | 3.00 | 4.50 | 3.00 |
| 14 | 3.00 | 4.00 | 4.50 | 3.50 |
| 15 | 3.50 | 5.00 | 4.50 | 3.00 |
| 16 | 3.00 | 7.00 | 4.50 | 0.50 |
| 17 | 0.50 | 8.00 | 4.50 | 0.00 |
| 18 | 0.00 | 4.00 | 4.50 | 0.50 |
| 19 | 0.50 | 3.00 | 4.50 | 2.00 |
| 20 | 2.00 | 3.00 | 4.50 | 3.50 |
| 21 | 3.50 | 2.00 | 4.50 | 6.00 |
| 22 | 6.00 | 1.00 | 4.50 | 9.50 |
| 23 | 9.50 | 3.00 | 4.50 | 11.00 |
| 24 | 11.00 | 6.00 | 4.50 | 9.50 |
| 25 | 9.50 | 8.00 | 4.50 | 6.00 |
| 26 | 6.00 | 9.00 | 4.50 | 1.50 |
| 27 | 1.50 | 3.00 | 4.50 | 3.00 |
| 28 | 3.00 | 4.00 | 4.50 | 3.50 |

THE REQUIRED STORAGE IS : 11.00