

Grades 9-12: Applications of Finite Mathematics

Adopted 2019

Applications of Finite Mathematics

Logical Reasoning

- A. The validity of a statement or argument can be determined using the models and language of first order logic. **FM.LR.A**
 - 1. Represent logic statements in words, with symbols, and in truth tables, including conditional, biconditional, converse, inverse, contrapositive, and quantified statements. **FM.LR.A.1**
 - 2. Represent logic operations such as **and**, **or**, **not**, **nor**, and **x** or (exclusive **or**) in words, with symbols, and in truth tables. **FM.LR.A.2**
 - 3. Use truth tables to solve application-based logic problems and determine the truth value of simple and compound statements including negations and implications. **FM.LR.A.3**
 - a. Determine whether statements are equivalent and construct equivalent statements. **FM.LR.A.3.A**
 - 4. Determine whether a logical argument is valid or invalid, using laws of logic such as the law of syllogism and the law of detachment. **FM.LR.A.4**
 - a. Determine whether a logical argument is a tautology or a contradiction. **FM.LR.A.4.A**
 - 5. Prove a statement indirectly by proving the contrapositive of the statement. **FM.LR.A.5**

Advanced Counting

- A. Complex counting problems can be solved efficiently using a variety of techniques. [FM.AC.A](#)
6. Use multiple representations and methods for counting objects and developing more efficient counting techniques. Note: Representations and methods may include tree diagrams, lists, manipulatives, overcounting methods, recursive patterns, and explicit formulas. [FM.AC.A.6](#)
 7. Develop and use the Fundamental Counting Principle for counting independent and dependent events. [FM.AC.A.7](#)
 - a. Use various counting models (including tree diagrams and lists) to identify the distinguishing factors of a context in which the Fundamental Counting Principle can be applied. [FM.AC.A.7.A](#)
 8. Using application-based problems, develop formulas for permutations, combinations, and combinations with repetition and compare student-derived formulas to standard representations of the formulas. [FM.AC.A.8](#)
 - a. Identify differences between applications of combinations and permutations. [FM.AC.A.8.A](#)
 - b. Using application-based problems, calculate the number of permutations of a set with n elements. Calculate the number of permutations of r elements taken from a set of n elements. [FM.AC.A.8.B](#)
 - c. Using application-based problems, calculate the number of subsets of size r that can be chosen from a set of n elements, explaining this number as the number of combinations " n choose r ."
[FM.AC.A.8.C](#)
 - d. Using application-based problems, calculate the number of combinations with repetitions of r elements from a set of n elements as " $(n + r - 1)$ choose r ."
[FM.AC.A.8.D](#)
 9. Use various counting techniques to determine probabilities of events. [FM.AC.A.9](#)
 10. Use the Pigeonhole Principle to solve counting problems. [FM.AC.A.10](#)

Recursion

- A. Recursion is a method of problem solving where a given relation or routine operation is repeatedly applied. [FM.R.A](#)
 - 11. Find patterns in application problems involving series and sequences, and develop recursive and explicit formulas as models to understand and describe sequential change. [FM.R.A.11](#)
 - 12. Determine characteristics of sequences, including the Fibonacci Sequence, the triangular numbers, and pentagonal numbers. [FM.R.A.12](#)
 - 13. Use the recursive process and difference equations to create fractals, population growth models, sequences, and series. [FM.R.A.13](#)
 - 14. Use mathematical induction to prove statements involving the positive integers. [FM.R.A.14](#)
 - 15. Develop and apply connections between Pascal's Triangle and combinations. [FM.R.A.15](#)

Networks

- A. Complex problems can be modeled using vertex and edge graphs and characteristics of the different structures are used to find solutions. **FM.N.A**
16. Use vertex and edge graphs to model mathematical situations involving networks. **FM.N.A.16**
 - a. Identify properties of simple graphs, complete graphs, bipartite graphs, complete bipartite graphs, and trees. **FM.N.A.16.A**
 17. Solve problems involving networks through investigation and application of existence and nonexistence of Euler paths, Euler circuits, Hamilton paths, and Hamilton circuits. **FM.N.A.17**
 - a. Develop optimal solutions of application-based problems using existing and student-created algorithms. **FM.N.A.17.A**
 - b. Give an argument for graph properties. **FM.N.A.17.B**
 18. Apply algorithms relating to minimum weight spanning trees, networks, flows, and Steiner trees. **FM.N.A.18**
 - a. Use shortest path techniques to find optimal shipping routes. **FM.N.A.18.A**
 - b. Show that every connected graph has a minimal spanning tree. **FM.N.A.18.B**
 - c. Use Kruskal's Algorithm and Prim's Algorithm to determine the minimal spanning tree of a weighted graph. **FM.N.A.18.C**
 19. Use vertex-coloring, edge-coloring, and matching techniques to solve application-based problems involving conflict. **FM.N.A.19**
 20. Determine the minimum time to complete a project using algorithms to schedule tasks in order, including critical path analysis, the list-processing algorithm, and student-created algorithms. **FM.N.A.20**
 21. Use the adjacency matrix of a graph to determine the number of walks of length n in a graph. **FM.N.A.21**

Fairness and Democracy

- A. Various methods for determining a winner in a voting system can result in paradoxes or other issues of fairness. [FM.FD.A](#)
- 22. Analyze advantages and disadvantages of different types of ballot voting systems. [FM.FD.A.22](#)
 - a. Identify impacts of using a preferential ballot voting system and compare it to single candidate voting and other voting systems. [FM.FD.A.22.A](#)
 - b. Analyze the impact of legal and cultural features of political systems on the mathematical aspects of elections. [FM.FD.A.22.B](#)
- 23. Apply a variety of methods for determining a winner using a preferential ballot voting system, including plurality, majority, run-off with majority, sequential run-off with majority, Borda count, pairwise comparison, Condorcet, and approval voting. [FM.FD.A.23](#)
- 24. Identify issues of fairness for different methods of determining a winner using a preferential voting ballot and other voting systems and identify paradoxes that can result. [FM.FD.A.24](#)
- 25. Use methods of weighted voting and identify issues of fairness related to weighted voting. Example: determine the power of voting bodies using the Banzhaf power index [FM.FD.A.25](#)
 - a. Distinguish between weight and power in voting. [FM.FD.A.25.A](#)

Fair Division

- A. Methods used to solve non-trivial problems of division of objects often reveal issues of fairness. [FM.FDV.A](#)
- 26. Explain and apply mathematical aspects of fair division, with respect to classic problems of apportionment, cake cutting, and estate division. Include applications in other contexts and modern situations. [FM.FDV.A.26](#)
- 27. Identify and apply historic methods of apportionment for voting districts including Hamilton, Jefferson, Adams, Webster, and Huntington-Hill. Identify issues of fairness and paradoxes that may result from methods. [FM.FDV.A.27](#)
- 28. Use spreadsheets to examine apportionment methods in large problems. [FM.FDV.A.28](#)

Information Processing

- A. Effective systems for sending and receiving information include components that impact accuracy, efficiency, and security. **FM.IP.A**
- 29. Critically analyze issues related to information processing including accuracy, efficiency, and security. **FM.IP.A.29**
- 30. Apply ciphers (encryption and decryption algorithms) and cryptosystems for encrypting and decrypting including symmetric-key or public-key systems. **FM.IP.A.30**
 - a. Use modular arithmetic to apply RSA (Rivest-Shamir-Adleman) public-key cryptosystems. **FM.IP.A.30.A**
 - b. Use matrices and their inverses to encode and decode messages. **FM.IP.A.30.B**
- 31. Apply error-detecting codes and error-correcting codes to determine accuracy of information processing. **FM.IP.A.31**
- 32. Apply methods of data compression. **FM.IP.A.32**