

Метод сжатия по Хаффману. Код Хэмминга.

Кубасов В.Ю., 1132249516

11 сентября 2024

НФИмд-01-24

Введение

Актуальность

- Кодирование информации используется повсеместно
- Исследования в поисках лучшего метода в узкой области и в общем спектре до сих пор ведутся

Цель работы:

- Исследовать актуальные и основные алгоритмы кодирования информации

Задачи работы:

- Изучить алгоритм Хаффмана для сжатия информации - Изучить алгоритм Хэминга для формирования самопроверяющегося кода

Классический алгоритм Хаффмана

Классический алгоритм Хаффмана

1. Итерируемся по всему тексту, подсчитывая количество раз, которое определённый символ нам встретился.
2. Формируем список символов, сортируя его по количеству каждого символа в порядке возрастания - это наши будущие листья бинарного дерева.
3. Начинаем строить дерево. На каждой итерации выбираем 2 узла с наименьшим значением (частотой). Для них создаём родительский узел, значением которого выступает сумма значений дочерних узлов. Сами дочерние узлы из нашего списка удаляем, а вместо них вносим только что сформированный узел.
4. Повторяем предыдущий пункт, пока в списке не останется всего лишь один узел (корень). На этом моменте дерево сформировано. Таким образом, путь до самых частовстречающихся символов будет самым коротким, а, следовательно, будет иметь наименьший код.

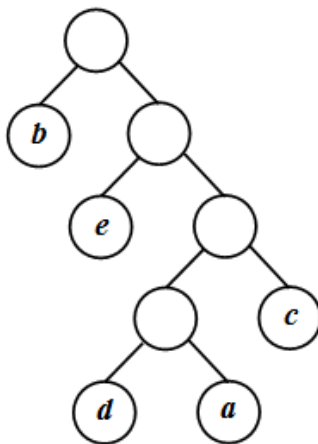
Классический алгоритм Хаффмана

Классический алгоритм Хаффмана

Вероятности появления символов

Символ	Вероятность
<i>a</i>	0,1
<i>b</i>	0,4
<i>c</i>	0,2
<i>d</i>	0,05
<i>e</i>	0,25

Кодовое дерево



Оптимальные префиксные коды

Символ	Код
<i>a</i>	1101
<i>b</i>	0
<i>c</i>	111
<i>d</i>	1100
<i>e</i>	10

Рис. 1: построение и алгоритм Хаффмана

Алгоритм Хэмминга

. Пусть исходное сообщение - 16 бит. 2. На каждую позицию, с порядковым номером, равным степени двойки вставляет контрольный бит (пока что равный нулю). 3. Каждый такой контрольный бит с порядковым номером N (начиная с 1) отвечает за следующие N символов через каждые N символов начиная с N .

На рисунке контрольные биты выделены красным, крестиками выделены биты, за которые они отвечают. 4. Далее просчитываем значение контрольного бита: если в его области ответственности нечетное количество единиц, то контрольный бит - единица. 5. На принимающей стороне контрольные биты заново высчитываются. Если контрольные биты не совпадают - ищем ошибку. Найти достаточно просто: нужно сложить порядковые номера всех бит, которые не совпали. Это и будет “битый” бит.

Алгоритм Хэмминга

Алгоритм Хэмминга

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	
X		X		X		X		X		X		X		X		X		X		X	1
	X	X			X	X			X	X			X	X			X	X			2
			X	X	X	X					X	X	X	X					X	X	4
							X	X	X	X	X	X	X	X							8
															X	X	X	X	X	X	16

Рис. 2: области ответственности контрольных битов

Выводы

1. Разобрали основные алгоритмы кодирования информации
2. Исследовали необходимость использования кодирования в современном мире