

## Code

```
1 from operator import gt, eq, concat
2 from itertools import count
3 from functools import reduce
4
5 from numpy import log2
6 from numpy import full as full_block
7 from numpy import empty as empty_block
8 from toolz import take, curry, compose, first, second
9
10
11 gt = curry(gt)
12 eq = curry(eq)
13
14
15 def take_while(predicate, iterable):
16     for element in iterable:
17         if predicate(element):
18             yield element
19         else:
20             break
21
22
23 def take_one_above(predicate, iterable):
24     for element in iterable:
25         if not predicate(element):
26             return element
27
28
29 def smaller_unsigned_integer_that_fits_(n):
30     return "uint" + str(
31         2 ** take_one_above(lambda i: 2**(2*i) - 1 < n, count(start=3))
32     )
33
34 class HashMap:
35     """A reimplementation of dictionaries using numpy arrays."""
36
```

```

37     def __init__(self, *args, **kwargs):
38         """Multiple Dispatch"""
39         keys, values = self.parse_input(*args, **kwargs)
40         self.allocate_memory(keys, values)
41
42     def parse_input(self, *args, **kwargs):
43         keys: list
44         values: list
45         if len(args) > 0:
46             if isinstance(pairs := args[0], list):
47                 if isinstance(pairs[0], tuple) and len(pairs[0]) == 2:
48                     keys, values = zip(*args[0])
49                 else:
50                     raise ValueError(f"Invalid input format.")
51             elif isinstance(string := args[0], str):
52                 keys, values = self.parse(string)
53             elif isinstance(dict_ := args[0], dict):
54                 keys, values = zip(*dict_.items())
55             else:
56                 raise ValueError(f"Invalid input format.")
57         elif len(kwargs) > 0:
58             keys, values = zip(*kwargs.items())
59         else:
60             raise ValueError(f"Invalid input format.")
61         return keys, values
62
63     @staticmethod
64     def parse(string: str):
65         string = string.strip()
66         if "{" not in string:
67             string = "{" + string + "}"
68         if "=>" in string:
69             string = string.replace("=>", ":")
70
71         d = eval(string)
72
73         return zip(*d.items())
74

```

```

75     def allocate_memory(self, keys, values):
76         size = len(keys)
77         self.number_of_slots = 3*size if size > 5 else 15
78
79         data_type = smaller_unsigned_integer_that_fits_(self.number_of_slots)
80         self.placeholder = 2**int(data_type[4:]) - 1
81         self.hash_array = full_block(self.number_of_slots, self.placeholder, dtype=ob)
82         self.pairs_array = empty_block(self.number_of_slots * 2 // 3 + 1, dtype=ob)
83
84         self.insert(keys, values)
85
86     def insert(self, keys, values):
87         for i, (key, value) in enumerate(zip(keys, values)):
88             index = hash(key) % self.number_of_slots
89
90             key_in_array = self.hash_array[index]
91             if key_in_array == self.placeholder:
92                 self.hash_array[index] = i
93             else:
94                 i = key_in_array
95
96             pair_in_array = self.pairs_array[i]
97             if pair_in_array is None:
98                 self.pairs_array[i] = (key, value)
99             elif isinstance(pair_in_array, tuple):
100                 self.pairs_array[i] = [pair_in_array, (key, value)]
101             elif not any(p[0] == key for p in pair_in_array):
102                 self.pairs_array[i].append((key, value))
103
104     def __str__(self):
105         return reduce(concat, (f"# {repr(key)} => {repr(value)},\n" for key, value in self.pairs_array))
106
107     def keys(self):
108         for pair in self.pairs_array:
109             if pair is None:
110                 continue
111             elif isinstance(pair, tuple):
112                 yield pair[0]

```

```

113         else:
114             yield from map(first, pair)
115
116     def values(self):
117         for pair in self.pairs_array:
118             if pair is None:
119                 continue
120             elif isinstance(pair, tuple):
121                 yield pair[1]
122             else:
123                 yield from map(second, pair)
124
125     def items(self):
126         for pair in self.pairs_array:
127             if pair is None:
128                 continue
129             elif isinstance(pair, tuple):
130                 yield pair
131             else:
132                 yield from pair
133
134     def display_internals(self):
135         print(f"HashMap:
136 =====
137 {self}
138
139 --> Hash Array <--
140 {self.hash_array}
141
142 --> Pairs Array <--
143 {self.pairs_array}
144 -----
145 """.replace(x := str(self.placeholder), len(x) * "_"))
146
147     def __getitem__(self, key):
148         hash_index = hash(key) % self.number_of_slots
149         pair_index = self.hash_array[hash_index]
150

```

```

151         pair = self.pairs_array[pair_index]
152         if isinstance(pair, tuple):
153             return pair[1]
154         elif isinstance(pair, list):
155             return next(p[1] for p in pair if p[0] == key)
156         else:
157             raise KeyError("Map key not found")
158
159
160     hashmap = HashMap("""
161         x => x*x for x in range(1, 80)
162     """)
163
164     hashmap.display_internals()
165     print(hashmap[25])

```

## Bytecode

line	byte	opname	arg	argval
1	0	LOAD_CONST	0	(0)
	2	LOAD_CONST	1	(( 'gt', 'eq', 'concat' ))
	4	IMPORT_NAME	0	(operator)
	6	IMPORT_FROM	1	(gt)
	8	STORE_NAME	1	(gt)
	10	IMPORT_FROM	2	(eq)
	12	STORE_NAME	2	(eq)
	14	IMPORT_FROM	3	(concat)
	16	STORE_NAME	3	(concat)
	18	POP_TOP		
2	20	LOAD_CONST	0	(0)
	22	LOAD_CONST	2	(( 'count', ))
	24	IMPORT_NAME	4	(itertools)
	26	IMPORT_FROM	5	(count)
	28	STORE_NAME	5	(count)
	30	POP_TOP		
3	32	LOAD_CONST	0	(0)

	34	LOAD_CONST	3	((('reduce',))
	36	IMPORT_NAME	6	(functools)
	38	IMPORT_FROM	7	(reduce)
	40	STORE_NAME	7	(reduce)
	42	POP_TOP		
5	44	LOAD_CONST	0	(0)
	46	LOAD_CONST	4	((('log2',))
	48	IMPORT_NAME	8	(numpy)
	50	IMPORT_FROM	9	(log2)
	52	STORE_NAME	9	(log2)
	54	POP_TOP		
6	56	LOAD_CONST	0	(0)
	58	LOAD_CONST	5	((('full',))
	60	IMPORT_NAME	8	(numpy)
	62	IMPORT_FROM	10	(full)
	64	STORE_NAME	11	(full_block)
	66	POP_TOP		
7	68	LOAD_CONST	0	(0)
	70	LOAD_CONST	6	((('empty',))
	72	IMPORT_NAME	8	(numpy)
	74	IMPORT_FROM	12	(empty)
	76	STORE_NAME	13	(empty_block)
	78	POP_TOP		
8	80	LOAD_CONST	0	(0)
	82	LOAD_CONST	7	((('take', 'curry', 'compose', 'first',
	84	IMPORT_NAME	14	(toolz)
	86	IMPORT_FROM	15	(take)
	88	STORE_NAME	15	(take)
	90	IMPORT_FROM	16	(curry)
	92	STORE_NAME	16	(curry)
	94	IMPORT_FROM	17	(compose)
	96	STORE_NAME	17	(compose)
	98	IMPORT_FROM	18	(first)
	100	STORE_NAME	18	(first)

	102	IMPORT_FROM	19	(second)
	104	STORE_NAME	19	(second)
	106	POP_TOP		
11	108	LOAD_NAME	16	(curry)
	110	LOAD_NAME	1	(gt)
	112	CALL_FUNCTION	1	
	114	STORE_NAME	1	(gt)
12	116	LOAD_NAME	16	(curry)
	118	LOAD_NAME	2	(eq)
	120	CALL_FUNCTION	1	
	122	STORE_NAME	2	(eq)
15	124	LOAD_CONST	8	(<code object take_while at 0x7f8d2d051
	126	LOAD_CONST	9	('take_while')
	128	MAKE_FUNCTION	0	
	130	STORE_NAME	20	(take_while)
23	132	LOAD_CONST	10	(<code object take_one_above at 0x7f8d2
	134	LOAD_CONST	11	('take_one_above')
	136	MAKE_FUNCTION	0	
	138	STORE_NAME	21	(take_one_above)
29	140	LOAD_CONST	12	(<code object smaller_unsigned_integer_
	142	LOAD_CONST	13	('smaller_unsigned_integer_that_fits_')
	144	MAKE_FUNCTION	0	
	146	STORE_NAME	22	(smaller_unsigned_integer_that_fits_)
34	148	LOAD_BUILD_CLASS		
	150	LOAD_CONST	14	(<code object HashMap at 0x7f8d2d057500
	152	LOAD_CONST	15	('HashMap')
	154	MAKE_FUNCTION	0	
	156	LOAD_CONST	15	('HashMap')
	158	CALL_FUNCTION	2	
	160	STORE_NAME	23	(HashMap)
160	162	LOAD_NAME	23	(HashMap)

	164	LOAD_CONST	16	('\\n x => x*x for x in range(1, 80))
		CALL_FUNCTION	1	
		STORE_NAME	24	(hashmap)
164		170 LOAD_NAME	24	(hashmap)
		172 LOAD_METHOD	25	(display_internals)
		174 CALL_METHOD	0	
		176 POP_TOP		
165		178 LOAD_NAME	26	(print)
		180 LOAD_NAME	24	(hashmap)
		182 LOAD_CONST	17	(25)
		184 BINARY_SUBSCR		
		186 CALL_FUNCTION	1	
		188 POP_TOP		
		190 LOAD_CONST	18	(None)
		192 RETURN_VALUE		

Disassembly of <code object take\_while at 0x7f8d2d0513a0, file "<dis>", line 15>:

16		0	LOAD_FAST	1	(iterable)
		2	GET_ITER		
	>>	4	FOR_ITER	24	(to 30)
		6	STORE_FAST	2	(element)
17		8	LOAD_FAST	0	(predicate)
		10	LOAD_FAST	2	(element)
		12	CALL_FUNCTION	1	
		14	POP_JUMP_IF_FALSE	24	
18		16	LOAD_FAST	2	(element)
		18	YIELD_VALUE		
		20	POP_TOP		
		22	JUMP_ABSOLUTE	4	
20	>>	24	POP_TOP		
		26	JUMP_ABSOLUTE	30	
		28	JUMP_ABSOLUTE	4	
	>>	30	LOAD_CONST	0	(None)



### 32 RETURN\_VALUE

Disassembly of <code object take\_one\_above at 0x7f8d2d051450, file "<dis>", line 2

```

24      0 LOAD_FAST          1 (iterable)
      2 GET_ITER
    >>  4 FOR_ITER              20 (to 26)
      6 STORE_FAST           2 (element)

25      8 LOAD_FAST           0 (predicate)
     10 LOAD_FAST           2 (element)
     12 CALL_FUNCTION        1
     14 POP_JUMP_IF_TRUE     4

26     16 LOAD_FAST           2 (element)
     18 ROT_TWO
     20 POP_TOP
     22 RETURN_VALUE
     24 JUMP_ABSOLUTE         4
    >>  26 LOAD_CONST         0 (None)
     28 RETURN_VALUE

```

Disassembly of <code object smaller\_unsigned\_integer\_that\_fits\_ at 0x7f8d2d051870,

```

30      0 LOAD_CONST          1 ('uint')
      2 LOAD_GLOBAL           0 (str)

31      4 LOAD_CONST           2 (2)
      6 LOAD_GLOBAL           1 (take_one_above)
      8 LOAD_CLOSURE         0 (n)
     10 BUILD_TUPLE           1
     12 LOAD_CONST            3 (<code object <lambda> at 0x7f8d2d0517c0)
     14 LOAD_CONST            4 ('smaller_unsigned_integer_that_fits_.<
     16 MAKE_FUNCTION         8 (closure)
     18 LOAD_GLOBAL           2 (count)
     20 LOAD_CONST            5 (3)
     22 LOAD_CONST            6 (('start',))
     24 CALL_FUNCTION_KW       1
     26 CALL_FUNCTION         2
     28 BINARY_POWER

```

30	30 CALL_FUNCTION	1
	32 BINARY_ADD	
	34 RETURN_VALUE	

Disassembly of <code object <lambda> at 0x7f8d2d0517c0, file "<dis>", line 31>:

31	0 LOAD_CONST	1 (2)
	2 LOAD_CONST	1 (2)
	4 LOAD_FAST	0 (i)
	6 BINARY_POWER	
	8 BINARY_POWER	
	10 LOAD_CONST	2 (1)
	12 BINARY_SUBTRACT	
	14 LOAD_DEREF	0 (n)
	16 COMPARE_OP	0 (<)
	18 RETURN_VALUE	

Disassembly of <code object HashMap at 0x7f8d2d057500, file "<dis>", line 34>:

34	0 LOAD_NAME	0 (__name__)
	2 STORE_NAME	1 (__module__)
	4 LOAD_CONST	0 ('HashMap')
	6 STORE_NAME	2 (__qualname__)
35	8 LOAD_CONST	1 ('A reimplementatation of dictionaries us
	10 STORE_NAME	3 (__doc__)
37	12 LOAD_CONST	2 (<code object __init__ at 0x7f8d2d05192
	14 LOAD_CONST	3 ('HashMap.__init__')
	16 MAKE_FUNCTION	0
	18 STORE_NAME	4 (__init__)
42	20 LOAD_CONST	4 (<code object parse_input at 0x7f8d2d05
	22 LOAD_CONST	5 ('HashMap.parse_input')
	24 MAKE_FUNCTION	0
	26 STORE_NAME	5 (parse_input)
63	28 LOAD_NAME	6 (staticmethod)

64	30 LOAD_NAME	7 (str)
	32 LOAD_CONST	6 (('string',))
	34 BUILD_CONST_KEY_MAP	1
	36 LOAD_CONST	7 (<code object parse at 0x7f8d2d051a80,
	38 LOAD_CONST	8 ('HashMap.parse')
	40 MAKE_FUNCTION	4 (annotations)
	42 CALL_FUNCTION	1
	44 STORE_NAME	8 (parse)
75	46 LOAD_CONST	9 (<code object allocate_memory at 0x7f8d2d051b00,
	48 LOAD_CONST	10 ('HashMap.allocate_memory')
	50 MAKE_FUNCTION	0
	52 STORE_NAME	9 (allocate_memory)
86	54 LOAD_CONST	11 (<code object insert at 0x7f8d2d051d40,
	56 LOAD_CONST	12 ('HashMap.insert')
	58 MAKE_FUNCTION	0
	60 STORE_NAME	10 (insert)
104	62 LOAD_CONST	13 (<code object __str__ at 0x7f8d2d051ea0,
	64 LOAD_CONST	14 ('HashMap.__str__')
	66 MAKE_FUNCTION	0
	68 STORE_NAME	11 (__str__)
107	70 LOAD_CONST	15 (<code object keys at 0x7f8d2d051f50, f
	72 LOAD_CONST	16 ('HashMap.keys')
	74 MAKE_FUNCTION	0
	76 STORE_NAME	12 (keys)
116	78 LOAD_CONST	17 (<code object values at 0x7f8d2d057030,
	80 LOAD_CONST	18 ('HashMap.values')
	82 MAKE_FUNCTION	0
	84 STORE_NAME	13 (values)
125	86 LOAD_CONST	19 (<code object items at 0x7f8d2d0570e0,
	88 LOAD_CONST	20 ('HashMap.items')
	90 MAKE_FUNCTION	0
	92 STORE_NAME	14 (items)

134	94 LOAD_CONST	21 (<code object display_internals at 0x7f8d2d051920, file "<dis>", line 37>)
	96 LOAD_CONST	22 ('HashMap.display_internals')
	98 MAKE_FUNCTION	0
	100 STORE_NAME	15 (display_internals)
147	102 LOAD_CONST	23 (<code object __getitem__ at 0x7f8d2d051920, file "<dis>", line 37>)
	104 LOAD_CONST	24 ('HashMap.__getitem__')
	106 MAKE_FUNCTION	0
	108 STORE_NAME	16 (__getitem__)
	110 LOAD_CONST	25 (None)
	112 RETURN_VALUE	

Disassembly of <code object \_\_init\_\_ at 0x7f8d2d051920, file "<dis>", line 37>:

39	0 LOAD_FAST	0 (self)
	2 LOAD_ATTR	0 (parse_input)
	4 LOAD_FAST	1 (args)
	6 LOAD_FAST	2 (kwargs)
	8 CALL_FUNCTION_EX	1
	10 UNPACK_SEQUENCE	2
	12 STORE_FAST	3 (keys)
	14 STORE_FAST	4 (values)
40	16 LOAD_FAST	0 (self)
	18 LOAD_METHOD	1 (allocate_memory)
	20 LOAD_FAST	3 (keys)
	22 LOAD_FAST	4 (values)
	24 CALL_METHOD	2
	26 POP_TOP	
	28 LOAD_CONST	1 (None)
	30 RETURN_VALUE	

Disassembly of <code object parse\_input at 0x7f8d2d0519d0, file "<dis>", line 42>:

45	0 LOAD_GLOBAL	0 (len)
	2 LOAD_FAST	1 (args)
	4 CALL_FUNCTION	1
	6 LOAD_CONST	1 (0)
	8 COMPARE_OP	4 (>)

	10 POP_JUMP_IF_FALSE	168
46	12 LOAD_GLOBAL	1 (isinstance)
	14 LOAD_FAST	1 (args)
	16 LOAD_CONST	1 (0)
	18 BINARY_SUBSCR	
	20 DUP_TOP	
	22 STORE_FAST	3 (pairs)
	24 LOAD_GLOBAL	2 (list)
	26 CALL_FUNCTION	2
	28 POP_JUMP_IF_FALSE	88
47	30 LOAD_GLOBAL	1 (isinstance)
	32 LOAD_FAST	3 (pairs)
	34 LOAD_CONST	1 (0)
	36 BINARY_SUBSCR	
	38 LOAD_GLOBAL	3 (tuple)
	40 CALL_FUNCTION	2
	42 POP_JUMP_IF_FALSE	78
	44 LOAD_GLOBAL	0 (len)
	46 LOAD_FAST	3 (pairs)
	48 LOAD_CONST	1 (0)
	50 BINARY_SUBSCR	
	52 CALL_FUNCTION	1
	54 LOAD_CONST	2 (2)
	56 COMPARE_OP	2 (==)
	58 POP_JUMP_IF_FALSE	78
48	60 LOAD_GLOBAL	4 (zip)
	62 LOAD_FAST	1 (args)
	64 LOAD_CONST	1 (0)
	66 BINARY_SUBSCR	
	68 CALL_FUNCTION_EX	0
	70 UNPACK_SEQUENCE	2
	72 STORE_FAST	4 (keys)
	74 STORE_FAST	5 (values)
	76 JUMP_ABSOLUTE	166

```

50    >> 78 LOAD_GLOBAL          5 (ValueError)
        80 LOAD_CONST            3 ('Invalid input format.')
```

82	CALL_FUNCTION	1
84	RAISE_VARARGS	1
86	JUMP_ABSOLUTE	206

  

```

51    >> 88 LOAD_GLOBAL          1 (isinstance)
        90 LOAD_FAST             1 (args)
        92 LOAD_CONST            1 (0)
        94 BINARY_SUBSCR
        96 DUP_TOP
        98 STORE_FAST            6 (string)
       100 LOAD_GLOBAL            6 (str)
       102 CALL_FUNCTION           2
       104 POP_JUMP_IF_FALSE      122
```

  

```

52    106 LOAD_FAST              0 (self)
       108 LOAD_METHOD            7 (parse)
       110 LOAD_FAST             6 (string)
       112 CALL_METHOD            1
       114 UNPACK_SEQUENCE         2
       116 STORE_FAST            4 (keys)
       118 STORE_FAST            5 (values)
       120 JUMP_ABSOLUTE         206
```

  

```

53    >> 122 LOAD_GLOBAL          1 (isinstance)
        124 LOAD_FAST            1 (args)
        126 LOAD_CONST           1 (0)
        128 BINARY_SUBSCR
        130 DUP_TOP
        132 STORE_FAST           7 (dict_)
        134 LOAD_GLOBAL          8 (dict)
        136 CALL_FUNCTION         2
        138 POP_JUMP_IF_FALSE    158
```

  

```

54    140 LOAD_GLOBAL            4 (zip)
        142 LOAD_FAST            7 (dict_)
        144 LOAD_METHOD           9 (items)
```

		146 CALL_METHOD	0
		148 CALL_FUNCTION_EX	0
		150 UNPACK_SEQUENCE	2
		152 STORE_FAST	4 (keys)
		154 STORE_FAST	5 (values)
		156 JUMP_ABSOLUTE	206
56	>>	158 LOAD_GLOBAL	5 (ValueError)
		160 LOAD_CONST	3 ('Invalid input format.')
		162 CALL_FUNCTION	1
		164 RAISE_VARARGS	1
	>>	166 JUMP_FORWARD	38 (to 206)
57	>>	168 LOAD_GLOBAL	0 (len)
		170 LOAD_FAST	2 (kwargs)
		172 CALL_FUNCTION	1
		174 LOAD_CONST	1 (0)
		176 COMPARE_OP	4 (>)
		178 POP_JUMP_IF_FALSE	198
58		180 LOAD_GLOBAL	4 (zip)
		182 LOAD_FAST	2 (kwargs)
		184 LOAD_METHOD	9 (items)
		186 CALL_METHOD	0
		188 CALL_FUNCTION_EX	0
		190 UNPACK_SEQUENCE	2
		192 STORE_FAST	4 (keys)
		194 STORE_FAST	5 (values)
		196 JUMP_FORWARD	8 (to 206)
60	>>	198 LOAD_GLOBAL	5 (ValueError)
		200 LOAD_CONST	3 ('Invalid input format.')
		202 CALL_FUNCTION	1
		204 RAISE_VARARGS	1
61	>>	206 LOAD_FAST	4 (keys)
		208 LOAD_FAST	5 (values)
		210 BUILD_TUPLE	2

## 212 RETURN\_VALUE

Disassembly of <code object parse at 0x7f8d2d051a80, file "<dis>", line 63>:

65		0 LOAD_FAST	0 (string)
		2 LOAD_METHOD	0 (strip)
		4 CALL_METHOD	0
		6 STORE_FAST	0 (string)
66		8 LOAD_CONST	1 ('{')
		10 LOAD_FAST	0 (string)
		12 COMPARE_OP	7 (not in)
		14 POP_JUMP_IF_FALSE	28
67		16 LOAD_CONST	1 ('{')
		18 LOAD_FAST	0 (string)
		20 BINARY_ADD	
		22 LOAD_CONST	2 ('}')
		24 BINARY_ADD	
		26 STORE_FAST	0 (string)
68	>>	28 LOAD_CONST	3 ('=>')
		30 LOAD_FAST	0 (string)
		32 COMPARE_OP	6 (in)
		34 POP_JUMP_IF_FALSE	48
69		36 LOAD_FAST	0 (string)
		38 LOAD_METHOD	1 (replace)
		40 LOAD_CONST	3 ('=>')
		42 LOAD_CONST	4 (':')
		44 CALL_METHOD	2
		46 STORE_FAST	0 (string)
71	>>	48 LOAD_GLOBAL	2 (eval)
		50 LOAD_FAST	0 (string)
		52 CALL_FUNCTION	1
		54 STORE_FAST	1 (d)
73		56 LOAD_GLOBAL	3 (zip)



58	LOAD_FAST	1 (d)
60	LOAD_METHOD	4 (items)
62	CALL_METHOD	0
64	CALL_FUNCTION_EX	0
66	RETURN_VALUE	

Disassembly of <code object allocate\_memory at 0x7f8d2d051c90, file "<dis>", line

76	0	LOAD_GLOBAL	0 (len)
	2	LOAD_FAST	1 (keys)
	4	CALL_FUNCTION	1
	6	STORE_FAST	3 (size)
77	8	LOAD_FAST	3 (size)
	10	LOAD_CONST	1 (5)
	12	COMPARE_OP	4 (>)
	14	POP_JUMP_IF_FALSE	24
	16	LOAD_CONST	2 (3)
	18	LOAD_FAST	3 (size)
	20	BINARY_MULTIPLY	
	22	JUMP_FORWARD	2 (to 26)
>>	24	LOAD_CONST	3 (15)
>>	26	LOAD_FAST	0 (self)
	28	STORE_ATTR	1 (number_of_slots)
79	30	LOAD_GLOBAL	2 (smaller_unsigned_integer_that_fits_)
	32	LOAD_FAST	0 (self)
	34	LOAD_ATTR	1 (number_of_slots)
	36	CALL_FUNCTION	1
	38	STORE_FAST	4 (data_type)
80	40	LOAD_CONST	4 (2)
	42	LOAD_GLOBAL	3 (int)
	44	LOAD_FAST	4 (data_type)
	46	LOAD_CONST	5 (4)
	48	LOAD_CONST	0 (None)
	50	BUILD_SLICE	2
	52	BINARY_SUBSCR	
	54	CALL_FUNCTION	1

	56	BINARY_POWER	
	58	LOAD_CONST	6 (1)
	60	BINARY_SUBTRACT	
	62	LOAD_FAST	0 (self)
	64	STORE_ATTR	4 (placeholder)
81	66	LOAD_GLOBAL	5 (full_block)
	68	LOAD_FAST	0 (self)
	70	LOAD_ATTR	1 (number_of_slots)
	72	LOAD_FAST	0 (self)
	74	LOAD_ATTR	4 (placeholder)
	76	LOAD_FAST	4 (data_type)
	78	LOAD_CONST	7 (('dtype',))
	80	CALL_FUNCTION_KW	3
	82	LOAD_FAST	0 (self)
	84	STORE_ATTR	6 (hash_array)
82	86	LOAD_GLOBAL	7 (empty_block)
	88	LOAD_FAST	0 (self)
	90	LOAD_ATTR	1 (number_of_slots)
	92	LOAD_CONST	4 (2)
	94	BINARY_MULTIPLY	
	96	LOAD_CONST	2 (3)
	98	BINARY_FLOOR_DIVIDE	
	100	LOAD_CONST	6 (1)
	102	BINARY_ADD	
	104	LOAD_GLOBAL	8 (object)
	106	LOAD_CONST	7 (('dtype',))
	108	CALL_FUNCTION_KW	2
	110	LOAD_FAST	0 (self)
	112	STORE_ATTR	9 (pairs_array)
84	114	LOAD_FAST	0 (self)
	116	LOAD_METHOD	10 (insert)
	118	LOAD_FAST	1 (keys)
	120	LOAD_FAST	2 (values)
	122	CALL_METHOD	2
	124	POP_TOP	

126	LOAD_CONST	0 (None)
128	RETURN_VALUE	

Disassembly of <code object insert at 0x7f8d2d051d40, file "<dis>", line 86>:

87	0	LOAD_GLOBAL	0 (enumerate)
	2	LOAD_GLOBAL	1 (zip)
	4	LOAD_FAST	1 (keys)
	6	LOAD_FAST	2 (values)
	8	CALL_FUNCTION	2
	10	CALL_FUNCTION	1
	12	GET_ITER	
>>	14	FOR_ITER	168 (to 184)
	16	UNPACK_SEQUENCE	2
	18	STORE_FAST	3 (i)
	20	UNPACK_SEQUENCE	2
	22	STORE_DEREF	0 (key)
	24	STORE_FAST	4 (value)
88	26	LOAD_GLOBAL	2 (hash)
	28	LOAD_DEREF	0 (key)
	30	CALL_FUNCTION	1
	32	LOAD_FAST	0 (self)
	34	LOAD_ATTR	3 (number_of_slots)
	36	BINARY_MODULO	
	38	STORE_FAST	5 (index)
90	40	LOAD_FAST	0 (self)
	42	LOAD_ATTR	4 (hash_array)
	44	LOAD_FAST	5 (index)
	46	BINARY_SUBSCR	
	48	STORE_FAST	6 (key_in_array)
91	50	LOAD_FAST	6 (key_in_array)
	52	LOAD_FAST	0 (self)
	54	LOAD_ATTR	5 (placeholder)
	56	COMPARE_OP	2 (==)
	58	POP_JUMP_IF_FALSE	72

92		60 LOAD_FAST	3 (i)
		62 LOAD_FAST	0 (self)
		64 LOAD_ATTR	4 (hash_array)
		66 LOAD_FAST	5 (index)
		68 STORE_SUBSCR	
		70 JUMP_FORWARD	4 (to 76)
94	>>	72 LOAD_FAST	6 (key_in_array)
		74 STORE_FAST	3 (i)
96	>>	76 LOAD_FAST	0 (self)
		78 LOAD_ATTR	6 (pairs_array)
		80 LOAD_FAST	3 (i)
		82 BINARY_SUBSCR	
		84 STORE_FAST	7 (pair_in_array)
97		86 LOAD_FAST	7 (pair_in_array)
		88 LOAD_CONST	0 (None)
		90 COMPARE_OP	8 (is)
		92 POP_JUMP_IF_FALSE	110
98		94 LOAD_DEREF	0 (key)
		96 LOAD_FAST	4 (value)
		98 BUILD_TUPLE	2
		100 LOAD_FAST	0 (self)
		102 LOAD_ATTR	6 (pairs_array)
		104 LOAD_FAST	3 (i)
		106 STORE_SUBSCR	
		108 JUMP_ABSOLUTE	14
99	>>	110 LOAD_GLOBAL	7 (isinstance)
		112 LOAD_FAST	7 (pair_in_array)
		114 LOAD_GLOBAL	8 (tuple)
		116 CALL_FUNCTION	2
		118 POP_JUMP_IF_FALSE	140
100		120 LOAD_FAST	7 (pair_in_array)
		122 LOAD_DEREF	0 (key)

```

124 LOAD_FAST          4 (value)
126 BUILD_TUPLE        2
128 BUILD_LIST         2
130 LOAD_FAST          0 (self)
132 LOAD_ATTR          6 (pairs_array)
134 LOAD_FAST          3 (i)
136 STORE_SUBSCR
138 JUMP_ABSOLUTE      14

101  >> 140 LOAD_GLOBAL      9 (any)
      142 LOAD_CLOSURE        0 (key)
      144 BUILD_TUPLE         1
      146 LOAD_CONST          1 (<code object <genexpr> at 0x7f8d2d051b30, file "<dis>", line 101>:
      148 LOAD_CONST          2 ('HashMap.insert.<locals>.<genexpr>')
      150 MAKE_FUNCTION       8 (closure)
      152 LOAD_FAST          7 (pair_in_array)
      154 GET_ITER
      156 CALL_FUNCTION         1
      158 CALL_FUNCTION         1
      160 POP_JUMP_IF_TRUE    14

102      162 LOAD_FAST          0 (self)
      164 LOAD_ATTR          6 (pairs_array)
      166 LOAD_FAST          3 (i)
      168 BINARY_SUBSCR
      170 LOAD_METHOD        10 (append)
      172 LOAD_DEREF         0 (key)
      174 LOAD_FAST          4 (value)
      176 BUILD_TUPLE        2
      178 CALL_METHOD         1
      180 POP_TOP
      182 JUMP_ABSOLUTE      14
      >> 184 LOAD_CONST          0 (None)
      186 RETURN_VALUE

Disassembly of <code object <genexpr> at 0x7f8d2d051b30, file "<dis>", line 101>:
101      0 LOAD_FAST          0 (.0)
      >>  2 FOR_ITER          18 (to 22)

```

	4	STORE_FAST	1	(p)
	6	LOAD_FAST	1	(p)
	8	LOAD_CONST	0	(0)
	10	BINARY_SUBSCR		
	12	LOAD_DEREF	0	(key)
	14	COMPARE_OP	2	(==)
	16	YIELD_VALUE		
	18	POP_TOP		
	20	JUMP_ABSOLUTE	2	
>>	22	LOAD_CONST	1	(None)
	24	RETURN_VALUE		

Disassembly of <code object \_\_str\_\_ at 0x7f8d2d051ea0, file "<dis>", line 104>:

105	0	LOAD_GLOBAL	0	(reduce)
	2	LOAD_GLOBAL	1	(concat)
	4	LOAD_CONST	1	(<code object <genexpr> at 0x7f8d2d051e...
	6	LOAD_CONST	2	('HashMap.__str__.<locals>.<genexpr>')
	8	MAKE_FUNCTION	0	
	10	LOAD_FAST	0	(self)
	12	LOAD_METHOD	2	(items)
	14	CALL_METHOD	0	
	16	GET_ITER		
	18	CALL_FUNCTION	1	
	20	CALL_FUNCTION	2	
	22	LOAD_CONST	0	(None)
	24	LOAD_CONST	3	(-2)
	26	BUILD_SLICE	2	
	28	BINARY_SUBSCR		
	30	RETURN_VALUE		

Disassembly of <code object <genexpr> at 0x7f8d2d051df0, file "<dis>", line 105>:

105	0	LOAD_FAST	0	(.0)
>>	2	FOR_ITER	36	(to 40)
	4	UNPACK_SEQUENCE	2	
	6	STORE_FAST	1	(key)
	8	STORE_FAST	2	(value)
	10	LOAD_CONST	0	(' # ')
	12	LOAD_GLOBAL	0	(repr)

	14	LOAD_FAST	1	(key)
	16	CALL_FUNCTION	1	
	18	FORMAT_VALUE	0	
	20	LOAD_CONST	1	(' => ')
	22	LOAD_GLOBAL	0	(repr)
	24	LOAD_FAST	2	(value)
	26	CALL_FUNCTION	1	
	28	FORMAT_VALUE	0	
	30	LOAD_CONST	2	(',\n')
	32	BUILD_STRING	5	
	34	YIELD_VALUE		
	36	POP_TOP		
	38	JUMP_ABSOLUTE	2	
>>	40	LOAD_CONST	3	(None)
	42	RETURN_VALUE		

Disassembly of <code object keys at 0x7f8d2d051f50, file "<dis>", line 107>:

108		0	LOAD_FAST	0	(self)
		2	LOAD_ATTR	0	(pairs_array)
		4	GET_ITER		
	>>	6	FOR_ITER	54	(to 62)
		8	STORE_FAST	1	(pair)
109		10	LOAD_FAST	1	(pair)
		12	LOAD_CONST	0	(None)
		14	COMPARE_OP	8	(is)
		16	POP_JUMP_IF_FALSE	22	
110		18	JUMP_ABSOLUTE	6	
		20	JUMP_ABSOLUTE	6	
111	>>	22	LOAD_GLOBAL	1	(isinstance)
		24	LOAD_FAST	1	(pair)
		26	LOAD_GLOBAL	2	(tuple)
		28	CALL_FUNCTION	2	
		30	POP_JUMP_IF_FALSE	44	
112		32	LOAD_FAST	1	(pair)

```

34 LOAD_CONST          1 (0)
36 BINARY_SUBSCR
38 YIELD_VALUE
40 POP_TOP
42 JUMP_ABSOLUTE       6

114  >> 44 LOAD_GLOBAL      3 (map)
      46 LOAD_GLOBAL      4 (first)
      48 LOAD_FAST         1 (pair)
      50 CALL_FUNCTION     2
      52 GET_YIELD_FROM_ITER
      54 LOAD_CONST        0 (None)
      56 YIELD_FROM
      58 POP_TOP
      60 JUMP_ABSOLUTE     6
      >> 62 LOAD_CONST        0 (None)
      64 RETURN_VALUE

```

Disassembly of <code object values at 0x7f8d2d057030, file "<dis>", line 116>:

```

117      0 LOAD_FAST       0 (self)
      2 LOAD_ATTR          0 (pairs_array)
      4 GET_ITER
      >> 6 FOR_ITER          54 (to 62)
      8 STORE_FAST        1 (pair)

118     10 LOAD_FAST        1 (pair)
      12 LOAD_CONST        0 (None)
      14 COMPARE_OP        8 (is)
      16 POP_JUMP_IF_FALSE 22

119     18 JUMP_ABSOLUTE     6
      20 JUMP_ABSOLUTE     6

120  >> 22 LOAD_GLOBAL      1 (isinstance)
      24 LOAD_FAST         1 (pair)
      26 LOAD_GLOBAL        2 (tuple)
      28 CALL_FUNCTION       2
      30 POP_JUMP_IF_FALSE 44

```



121		32 LOAD_FAST	1 (pair)
		34 LOAD_CONST	1 (1)
		36 BINARY_SUBSCR	
		38 YIELD_VALUE	
		40 POP_TOP	
		42 JUMP_ABSOLUTE	6
123	>>	44 LOAD_GLOBAL	3 (map)
		46 LOAD_GLOBAL	4 (second)
		48 LOAD_FAST	1 (pair)
		50 CALL_FUNCTION	2
		52 GET_YIELD_FROM_ITER	
		54 LOAD_CONST	0 (None)
		56 YIELD_FROM	
		58 POP_TOP	
		60 JUMP_ABSOLUTE	6
	>>	62 LOAD_CONST	0 (None)
		64 RETURN_VALUE	

Disassembly of <code object items at 0x7f8d2d0570e0, file "<dis>", line 125>:

126		0 LOAD_FAST	0 (self)
		2 LOAD_ATTR	0 (pairs_array)
		4 GET_ITER	
	>>	6 FOR_ITER	44 (to 52)
		8 STORE_FAST	1 (pair)
127		10 LOAD_FAST	1 (pair)
		12 LOAD_CONST	0 (None)
		14 COMPARE_OP	8 (is)
		16 POP_JUMP_IF_FALSE	22
128		18 JUMP_ABSOLUTE	6
		20 JUMP_ABSOLUTE	6
129	>>	22 LOAD_GLOBAL	1 (isinstance)
		24 LOAD_FAST	1 (pair)
		26 LOAD_GLOBAL	2 (tuple)

		28 CALL_FUNCTION	2
		30 POP_JUMP_IF_FALSE	40
130		32 LOAD_FAST	1 (pair)
		34 YIELD_VALUE	
		36 POP_TOP	
		38 JUMP_ABSOLUTE	6
132	>>	40 LOAD_FAST	1 (pair)
		42 GET_YIELD_FROM_ITER	
		44 LOAD_CONST	0 (None)
		46 YIELD_FROM	
		48 POP_TOP	
		50 JUMP_ABSOLUTE	6
	>>	52 LOAD_CONST	0 (None)
		54 RETURN_VALUE	

Disassembly of <code object display\_internals at 0x7f8d2d057190, file "<dis>", line 135

135		0 LOAD_GLOBAL	0 (print)
		2 LOAD_CONST	1 ('HashMap:\n=====\\n')
137		4 LOAD_FAST	0 (self)
135		6 FORMAT_VALUE	0
		8 LOAD_CONST	2 ('\\n\\n--> Hash Array <--\\n')
140		10 LOAD_FAST	0 (self)
		12 LOAD_ATTR	1 (hash_array)
135		14 FORMAT_VALUE	0
		16 LOAD_CONST	3 ('\\n\\n--> Pairs Array <--\\n')
143		18 LOAD_FAST	0 (self)
		20 LOAD_ATTR	2 (pairs_array)
135		22 FORMAT_VALUE	0
		24 LOAD_CONST	4 ('\\n-----\\n')
		26 BUILD_STRING	7

	28	LOAD_METHOD	3 (replace)
145	30	LOAD_GLOBAL	4 (str)
	32	LOAD_FAST	0 (self)
	34	LOAD_ATTR	5 (placeholder)
	36	CALL_FUNCTION	1
	38	DUP_TOP	
	40	STORE_FAST	1 (x)
	42	LOAD_GLOBAL	6 (len)
	44	LOAD_FAST	1 (x)
	46	CALL_FUNCTION	1
	48	LOAD_CONST	5 ('_')
	50	BINARY_MULTIPLY	
135	52	CALL_METHOD	2
	54	CALL_FUNCTION	1
	56	POP_TOP	
	58	LOAD_CONST	0 (None)
	60	RETURN_VALUE	

Disassembly of <code object \_\_getitem\_\_ at 0x7f8d2d0573a0, file "<dis>", line 147>

148	0	LOAD_GLOBAL	0 (hash)
	2	LOAD_DEREF	0 (key)
	4	CALL_FUNCTION	1
	6	LOAD_FAST	0 (self)
	8	LOAD_ATTR	1 (number_of_slots)
	10	BINARY_MODULO	
	12	STORE_FAST	2 (hash_index)
149	14	LOAD_FAST	0 (self)
	16	LOAD_ATTR	2 (hash_array)
	18	LOAD_FAST	2 (hash_index)
	20	BINARY_SUBSCR	
	22	STORE_FAST	3 (pair_index)
151	24	LOAD_FAST	0 (self)
	26	LOAD_ATTR	3 (pairs_array)
	28	LOAD_FAST	3 (pair_index)

		30 BINARY_SUBSCR	
		32 STORE_FAST	4 (pair)
152		34 LOAD_GLOBAL	4 (isinstance)
		36 LOAD_FAST	4 (pair)
		38 LOAD_GLOBAL	5 (tuple)
		40 CALL_FUNCTION	2
		42 POP_JUMP_IF_FALSE	52
153		44 LOAD_FAST	4 (pair)
		46 LOAD_CONST	1 (1)
		48 BINARY_SUBSCR	
		50 RETURN_VALUE	
154	>>	52 LOAD_GLOBAL	4 (isinstance)
		54 LOAD_FAST	4 (pair)
		56 LOAD_GLOBAL	6 (list)
		58 CALL_FUNCTION	2
		60 POP_JUMP_IF_FALSE	84
155		62 LOAD_GLOBAL	7 (next)
		64 LOAD_CLOSURE	0 (key)
		66 BUILD_TUPLE	1
		68 LOAD_CONST	2 (<code object <genexpr> at 0x7f8d2d0572
		70 LOAD_CONST	3 ('HashMap.__getitem__.<locals>.<genexpr
		72 MAKE_FUNCTION	8 (closure)
		74 LOAD_FAST	4 (pair)
		76 GET_ITER	
		78 CALL_FUNCTION	1
		80 CALL_FUNCTION	1
		82 RETURN_VALUE	
157	>>	84 LOAD_GLOBAL	8 (KeyError)
		86 LOAD_CONST	4 ('Map key not found')
		88 CALL_FUNCTION	1
		90 RAISE_VARARGS	1
		92 LOAD_CONST	0 (None)
		94 RETURN_VALUE	

```

Disassembly of <code object <genexpr> at 0x7f8d2d057240, file "<dis>", line 155>:
155      0 LOAD_FAST          0 (.0)
      >>  2 FOR_ITER              26 (to 30)
          4 STORE_FAST         1 (p)
          6 LOAD_FAST         1 (p)
          8 LOAD_CONST        0 (0)
         10 BINARY_SUBSCR
         12 LOAD_DEREF         0 (key)
         14 COMPARE_OP        2 (==)
         16 POP_JUMP_IF_FALSE  2
         18 LOAD_FAST         1 (p)
         20 LOAD_CONST        1 (1)
         22 BINARY_SUBSCR
         24 YIELD_VALUE
         26 POP_TOP
         28 JUMP_ABSOLUTE       2
      >>  30 LOAD_CONST        2 (None)
         32 RETURN_VALUE

```